UCRL-TR-217063

LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

A Numerical Study on X-Ray Diffraction Effects within Objects

Sean K. Lehman

November 14, 2005

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

This work was performed under the auspices of the U.S. Department of Energy by University of California, Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

A Numerical Study on X-Ray Diffraction Effects within Objects

Sean K. Lehman

November 14, 2005

UCRL-TR-217063

Abstract

X-rays, being waves, *always* undergo the propagation effects of reflection, refraction, diffraction, geometric attenuation and absorption. In most circumstances the first four effects are considered negligible given the resolution sizes demanded of the measurement systems, x-ray energies involved, and physical properties of the materials under evaluation. We have reached the point, however, in some x-ray non-destructive evaluation (NDE) and imaging where we wish to resolve features of *micrometer size* in *millimeter size* objects to less than *micrometer resolution*. Given this resolution and the sizes of the measurement systems, diffraction effects within the object may become observable. We studied the extent to which diffraction is observable numerically using a two-dimensional paraxial approximation wave propagation code using a multislice method. We modeled realistic parts of interest at worst-case x-ray energies, comparing wave propagation and straight-ray simulated results. In two cases, we compare the numerical results to experimental measurements. The conclusion, based upon the results of the simulation code, is that diffraction effects on the measured data will be insignificant. However, we demonstrate by a single example, that in certain cases diffraction effects may be significant.

Contents

1 Introduction

X-rays, being waves, *always* undergo the propagation effects of reflection, refraction, diffraction, geometric attenuation, and absorption. In most circumstances the first four effects are considered negligible given the resolution sizes of the measurement systems, x-ray energies involved, and physical properties of the materials under evaluation. We have assumed x-ray propagation is described by a *transport equation* and that we may adopt the particle interpretation of straight-ray propagation in which absorption only is taken into account.

We have reached the point, however, in x-ray non-destructive evaluation (NDE) and imaging where we wish to resolve features of *micrometer size* in *millimeter size* objects to less than *micrometer resolution*. Given this resolution and the sizes of the measurement systems, diffraction effects within the object may become observable and we must use a *wave equation*, the particle or transport model being incapable of accounting for diffraction.

In determining the transition from transport to wave equation, we have adopted a model developed by Gbur and Wolf [1] which describes the point at which diffraction effects become significant. The model is based upon the size of the feature to be resolved (in this case a micrometer) and the size of the measurement system; it ignores material properties such as density and was derived under the Born approximation which assumes small changes in refractive index. The relationship states [1],

$$L \ll \frac{\delta^2}{2\lambda}, \tag{1}$$

where L is the distance from the illuminated face of the object under evaluation to the detector, λ is the illuminating wavelength, and δ is the feature size. These dimensions are shown in Figure 1. The relationship states that if L is significantly larger than the ratio on the right hand side of Eqn. 1, then diffraction effects will be observed. One caveat we must consider in practical measurement systems is that the detector resolution may be insufficient to record these effects. Another is that, as stated, the formula was derived under the Born approximation and may not be applicable given the physical properties of some of the parts and materials we wish to evaluate.

We studied the extent to which diffraction is observable numerically using a two-dimensional paraxial approximation wave propagation code and a straight-ray propagation code. For the former, a multislice method is used to propagate the field within the object from entry to exit plane. For both the wave and straight-ray propagated models, a plane-to-plane Fourier method is used to propagate the field from the exit to the measurement plane. The code accepts a complex refractive index, thus accounting for absorption. The wave code naturally accounts for geometric attenuation because of its wave-based nature. The code does not explicitly model density unless it is accounted for in the complex refractive index. A rule-of-thumb in wave propagation and scattering codes is that the spatial sample interval must be finer than a tenth of a wavelength ($\Delta h \leq \lambda/10$). At x-ray wavelengths, this is not possible in millimeter size objects. For example, the wavelength at our lowest x-ray energy of 8 keV is 0.155 nm. The smallest spatial resolution we achieved using the paraxial approximation wave propagation code was 60 nm which is over 3.5 orders of magnitude greater than the minimum required for an accurate simulation. A consequence of this resolution

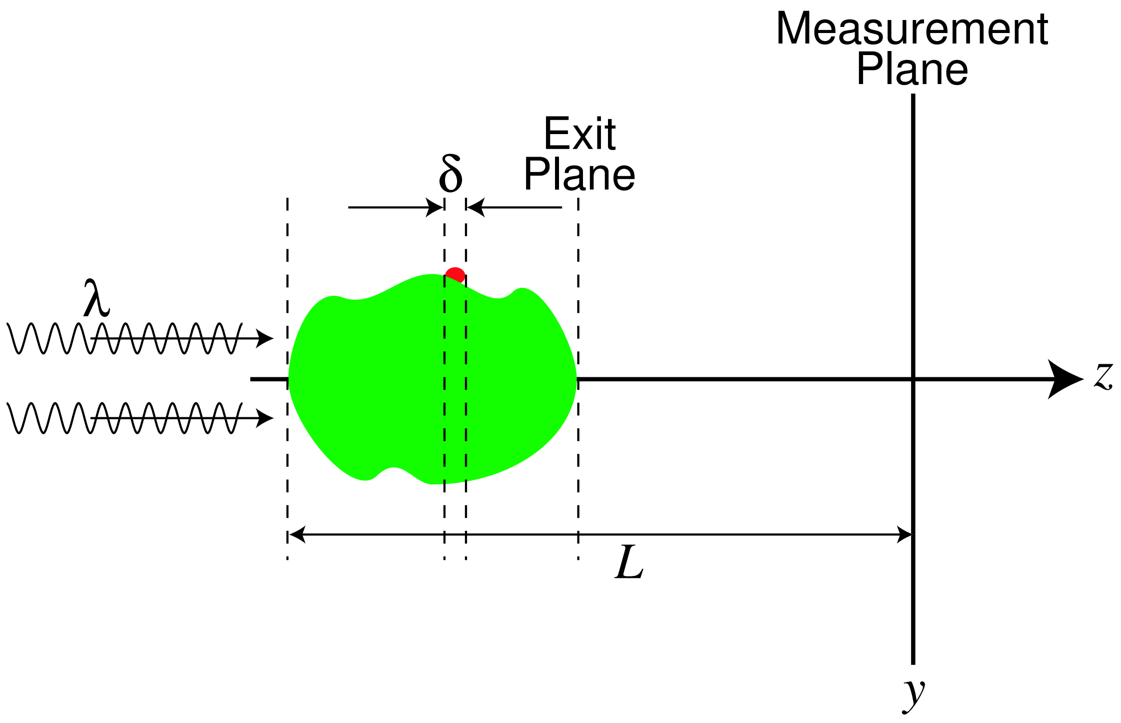


Figure 1: *Geometry of Gbur and Wolf. δ is the dimension of the smallest scattering feature to be resolved, L is the distance from the illuminated side of the object to the measurement plane. The field, with wavelength λ, is incident from the left.*

limitation is that the millimeter size features will be crudely pixelated or sampled with respect to the illuminating x-ray field. This effect is observable in all the results.

We modeled the following objects at a worst-case, that is low, x-ray energy (8 keV), comparing wave and straight-ray propagation:

- Two obscuring copper edges;
- A gold covered copper (Au/Cu) edge;
- An arrangement of three 5 μm diameter carbon fibers;
- A cylindrical multishell with and without 1 μm defects on its surface.

In parallel with the simulation, two experimental measurements were performed at 8 keV: one of the gold covered copper edge, another of the three 5 μm diameter carbon fibers. The experimental parameters are listed in Table 1. The refractive indices used in the simulations are listed in Appendix B.

The Au/Cu edge data were used to estimate the point spread function (PSF) (or rather, the *line spread function*) of the imaging system. Using the assumption the PSF has the form of either a single or sum of two Gaussians, a least mean squares (LMS) algorithm was used to fit the PSF and estimate the Gaussian parameter(s). The goal was to apply this PSF to the simulated carbon fiber data and verify the code by comparing them with the experimental carbon fiber data. We were not successful at this and demonstrate by fitting the experimental carbon fiber data to the simulated data, the PSFs of the edge and fiber data differ.

The goal of this study, however, was to determine the extent to which diffraction effects are observable. The conclusion, based upon the results of the simulation code, is that diffraction effects on the measured data will be insignificant, particularly given the blurring by the system PSF. However, we demonstrate by a single example, that in certain cases diffraction effects may be significant.

In the following section, we present the simulation of the two obscuring copper edges. They are arranged such that no straight-ray propagated field will pass and reach the measurement plane. Only the wave propagated diffracted field will pass.

Section 3, presents and discusses the measured and simulated gold covered copper edge data. The 5 μm diameter carbon fiber results are presented in Section 4. The cylindrical multishell simulation is treated in Section 5.

Sections 3 and 4 show single Gaussian PSFs. Double Gaussian PSF fits and results are summarized in Appendix A.

Tables in Appendix B list the material parameters used in the simulation. Appendix C lists the MATLAB `.m` files used to process and analyze the data.

2 Obscuring Copper Edges Test

As an initial test we modeled the two offset copper edges shown in profile in Figure 2. The field is incident from the left. The edges are offset along the propagation direction, z, but arranged

Au/Cu edge data	
sod	75 mm
odd	683 mm
sdd	758 mm
Magnification	$\frac{\mathrm{sdd}}{\mathrm{sod}} = 10.1$
Equivalent plane wave measurement plane	$\frac{\mathrm{odd}}{\mathrm{M}} = 67.62\ \mathrm{mm}$
Δy	2.47 μm

(a)

3/11/2005 Three Carbon Fiber Data	
sod	47.7 mm
odd	716 mm
sdd	763.7 mm
Magnification	$\frac{\mathrm{sdd}}{\mathrm{sod}} = 16.0$
Equivalent plane wave measurement plane	$\frac{\mathrm{odd}}{\mathrm{M}} = 44.72\ \mathrm{mm}$
Δy	1.56 μm

(b)

Table 1: *8 keV measurement parameters for the (a) Au/Cu edge, (b) 3/11/2005 three carbon fiber experimental data sets.*

Au/Cu Edge Data	
σ (μm)	2.96
FWHM (μm)	7.32
HM	0.02

03/11/2005 Carbon Fiber-Based PSFs		
	Double Fiber	**Single Fiber**
σ (μm)	4.38	3.97
FWHM (μm)	10.25	8.87
HM	0.02	0.02

Table 2: *Gaussian parameters for the PSF fits.*

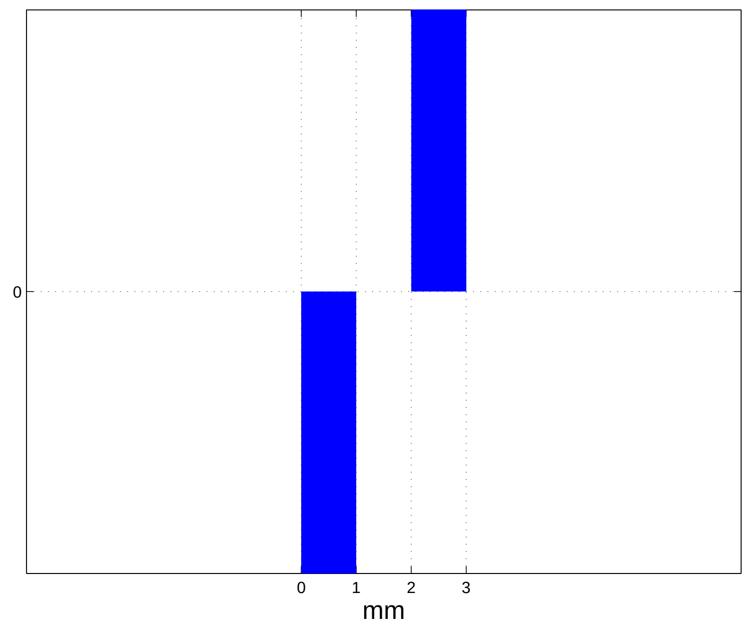


Figure 2: *Double obscuring copper edge test geometry. The field is incident from the left. The edges are offset along the propagation direction, z, but arranged vertically so that when viewed face-on they form a completely obscuring barrier to the incident field.*

vertically so that when viewed face-on they form a completely obscuring barrier to the incident field. With a material of sufficiently high attenuation and thickness, only the diffracted field, if any, should reach the detector. Any fields "propagated" by a transport equation will be stopped at the edges.

For the example presented here, we simulated 1 mm of copper illuminated by 8 keV x-rays ($\lambda = 0.155$ nm). Using the formula of Eqn. 1, we would expect no observable diffraction effects for measurement planes placed within a distance of

$$L = \frac{(10^{-3})^2}{2 \times 0.155 \times 10^{-9}} \approx 3.2 \text{ km}. \tag{2}$$

We compare the results of straight-ray and wave propagation in Figure 3. The received measured field intensity is shown for measurement planes placed at 1, 5, 10, and 100 cm. These are well within the 3.2 km limit, yet the attenuation and thickness is sufficient to annihilate all straight-ray propagation: the received field is purely diffractive. We emphasize, again, the formula of Eqn. 1 was derived under the Born approximation which is violated in this example. Additionally, it does not model material properties other than complex refractive index. This example does provide a verification that the multislice paraxial approximation wave propagated code can model diffraction around sharp edges.

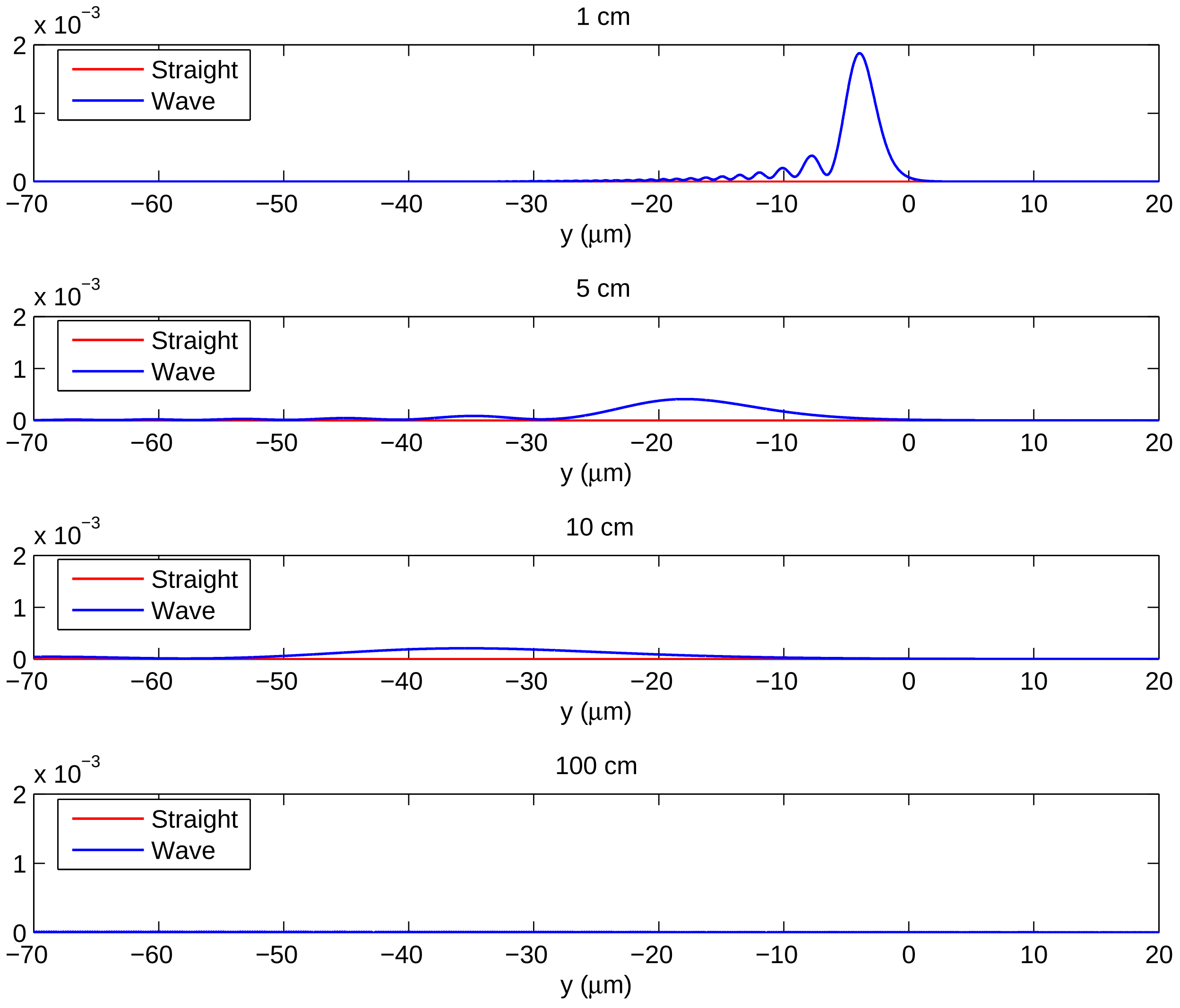


Figure 3: *Obscuring double edge results. From top to bottom, we plot the received field intensity at measurement planes placed at 1, 5, 10, and 100 cm. We simulated straight-ray (red trace) and wave (blue trace) propagation. The attenuation and thickness of the copper edges is sufficient to annihilate all straight-ray propagation; only the diffracted field is observed (measured).*

3 Gold Covered Copper Edge

We simulated 8 keV illumination of a gold covered copper surface with a radius of curvature of 37.5 mm. The gold layer is 60 μm thick. The part has a width of 3 mm. A graphic of the part is shown in Figure 4. The simulation models an experiment whose parameters are listed in Table 1(a).

The goal of the experiment was to estimate the point spread function (PSF), or more accurately, the *line spread function* (LSF) of the measurement system. Consider the following, ideal, two-dimensional edge object function described by

$$o(x,y) \;=\; \begin{cases} 0 & y < 0 \quad \forall x, \\ \alpha & y \geq 0 \quad \forall x, \end{cases} \tag{3}$$

where α is the amplitude of the edge. The image formed by the x-ray imaging system is the convolution of the object function with the PSF of the measurement system:

$$\begin{aligned} f(x,y) &= \int_{-\infty}^{\infty} dy' \int_{-\infty}^{\infty} dx' \; o(x',y') \; h(x-x', y-y'), && (4) \\ &= \alpha \int_{0}^{\infty} dy' \int_{-\infty}^{\infty} dx' \; h(x-x', y-y'), && (5) \end{aligned}$$

where we wish to determine $h(x,y)$.

To estimate the PSF from the edge response, we compute the derivative of $f(x,y)$ w.r.t. y:

$$\begin{aligned} \partial_y f(x,y) &= \alpha \int_{0}^{\infty} dy' \int_{-\infty}^{\infty} dx' \; \partial_y h(x-x', y-y'), && (6) \\ &= \alpha \int_{-\infty}^{\infty} dx' \; h(x-x', y-y')\Big|_{y'=0}^{y'=\infty}, && (7) \\ &= \alpha \int_{-\infty}^{\infty} dx' \; h(x-x', y), && (8) \end{aligned}$$

where we have used the Liebitz Derivative of an Integral and the assumption the PSF is zero at infinity. Note: this does not yield the PSF in y. Rather, it yields the PSF integrated over x. *If we assume the two-dimensional PSF is separable*, i.e. can be written as

$$h(x,y) \;=\; h_x(x) \; h_y(y), \tag{9}$$

then we have

$$\partial_y f(x,y) \;=\; \alpha \, h_y(y) \int_{-\infty}^{\infty} dx' \; h_x(x-x') \;=\; \alpha \, h_y(y), \tag{10}$$

with the assumption

$$\int_{-\infty}^{\infty} dx \; h_x(x) \;\equiv\; 1. \tag{11}$$

Note, if the PSF is not separable, then estimating the LSF from the Au/Cu edge data and applying it to the carbon fiber data is not valid. If it is separable but Eqn. 11 does not hold, then we should be off by a simple scaling factor. This does not appear to be the case.

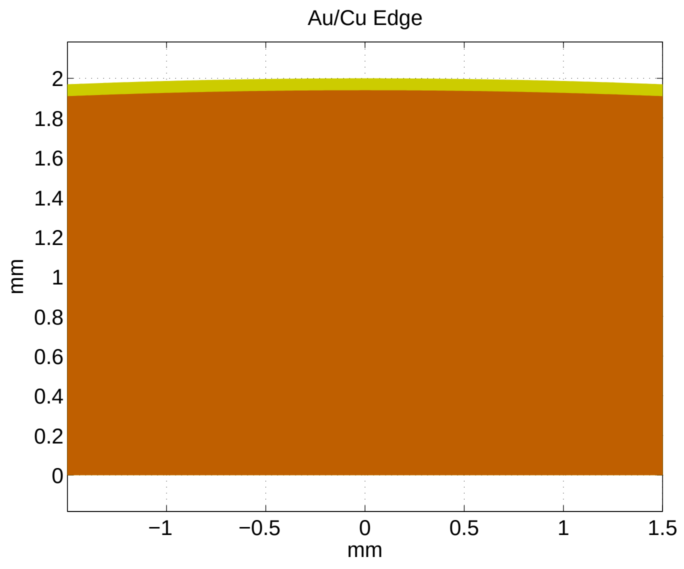


Figure 4: *Simulation geometry for gold covered copper edge.*

Figure 5 presents the results of the Au/Cu edge experiment and simulation. Figure 5(a) shows the measured data, simulated straight-ray transmission, and simulated wave propagated transmission. Using Eqn. 10, we computed the PSF from the measured data via a centered two-point difference. This is shown in Figure 5(b). In order to compare data sets (both measured and simulated) accurately, we had to resample them to a common sample interval (Δy). Figure 5(b) overlays the pre- and post-resampled measured Au/Cu edge data PSF. A Gaussian model for the PSF was adopted,

$$h(y,\sigma) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left[-\frac{1}{2}\left(\frac{y}{\sigma}\right)^2\right], \tag{12}$$

and fit to the measured data PSF using a least mean square algorithm[1] via

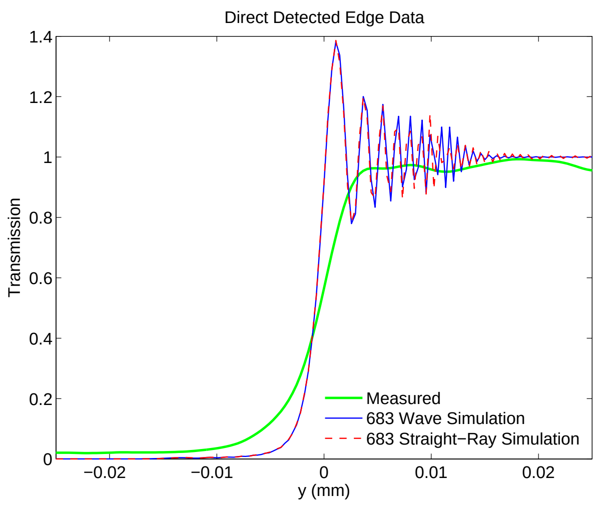
$$\hat{\sigma} = \underset{\sigma}{\operatorname{argmin}} \sum_n \left[h_y(y_n) - h(y_n,\sigma)\right]^2, \tag{13}$$

where y_n are the discrete y-axis sample points, and $h_y(y_n)$ is the PSF computed from the measured data via Eqn. 10 with the assumption of Eqn. 11. The fit returned a width of $\sigma = 2.96\mu$m. The results are listed in Table 2 and plotted in Figure 8(b).

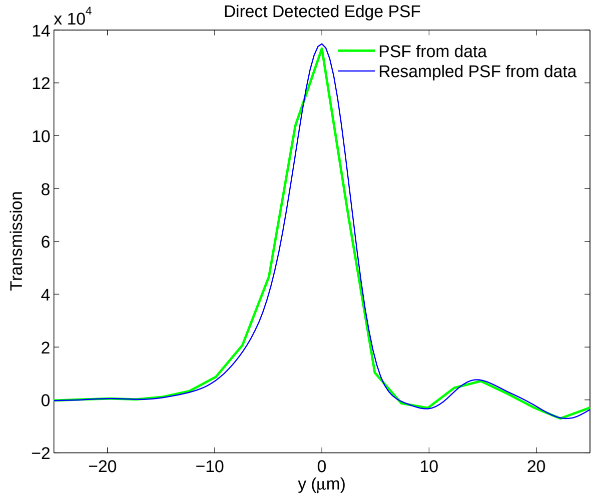
4 Three 5 μm Diameter Carbon Fiber

As a code validation test, we considered the configuration of three 5μm diameter carbon wires shown in Figure 6. With respect to the incident field, two of the wires are separated in the plane

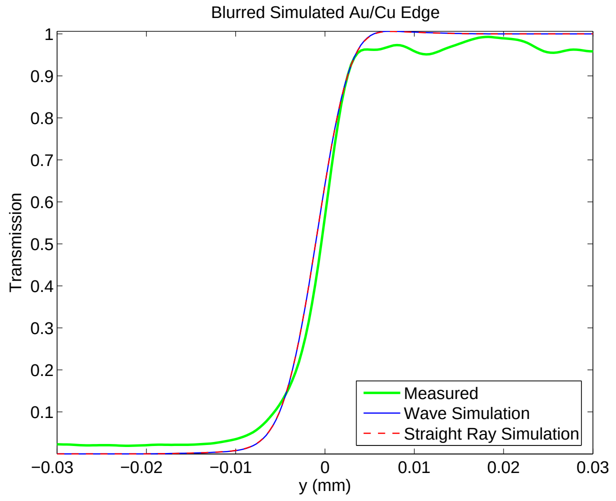
[1]We used MATLAB's `fminsearch` function.



(a)



(b)



(c)

Figure 5: *Comparison of measured and simulated Au/Cu edge results. (a) Superposition of measured transmission and simulated wave and straight-ray propagated transmissions for the experimental parameters of Table 1(a). (b) comparison of measured PSF and the resampled measured PSF. (c) the simulated data convolved with the* $\sigma = 2.96\mu$*m Gaussian fit to the measured PSF.*

perpendicular to the direction of propagation; while two are arranged so that one precisely optically shadows the other. We simulated wire a separation of 10.2 mm at 8 keV, and compared the results with experimental measurements on an actual arrangement of carbon wires using an 8 keV source.

The data are presented in Figure 7. The top two plots are the full resolution, that is pre-convolution, simulated results for the wave and straight-ray propagated fields. The bottom plots show the measured transmission data for the object. We observe there is minimal difference between the straight-ray and paraxial wave propagated fields. In order to model the detector element blur, the simulated intensities were convolved with a Gaussian of the form of Eqn. 12, where the width, σ, was determined by convolving the simulated results and comparing them with the actual data via

$$\hat{\sigma} \quad = \quad \underset{\sigma}{\operatorname{argmin}} \sum_{n} \left[d(y_n) - \operatorname{conv}(f(y_n), h(y_n, \sigma)) \right]^2 , \tag{14}$$

where $d(y_n)$ are the measured data at discrete sample location y_n, $f(y_n)$ are the simulated data, $h(y_n, \sigma)$ is the Gaussian of Eqn. 12, and $\operatorname{conv}(\cdot, \cdot)$ is the discrete convolution operator. In this manner, we arrived at values of 3.97μm for the single fiber, and 4.38μm for the double fiber. These Gaussians are presented in Figure 8 were we also compare the Gaussian fit to the Au/Cu edge measured PSF.

Figure 9(a) shows the simulated data convolved with the Au/Cu edge measured PSF; Figure 9(b) shows the data convolved with the PSF fit via Eqn. 14.

Our conclusion for this case, justified by the minimal difference in wave propagated and straight-ray simulated results, is that diffraction is not observable. This is particularly true when the data are blurred due to the large detector size.

5 Multishell Geometry

We modeled the cylindrical multishell described in Table 3 and shown graphically in Figure 10. We modeled the part under two configurations: one consisting of a smooth symmetric multishell, and another in which three "defects" in the form of 1 μm bumps placed on the outer surface at 90, 135, and 180 degrees. Additionally, we simulated the multishell at two energies: 8 keV ($\lambda = 0.155$ nm) and 30 keV ($\lambda = 0.0414$ nm).

The results presented in Figure 11 for both energies show obvious differences in the model without the "defects" and that with it. However, very little difference is seen between the straight-ray and wave propagated measured fields. Following the formula of Eqn. 1, we expect to observe diffraction effects after 3.2 mm at 8 keV and 12 meters at 30 keV for a 1 μm defect. We conclude that for this particular combination of geometry, sizes, and material properties, that diffraction can be neglected.

For completeness, we show in Figures 12 the results of Figure 11 with a Gaussian PSF of 2.96 μm applied to the measured field intensities. Under these conditions, no diffraction effects will be observed.

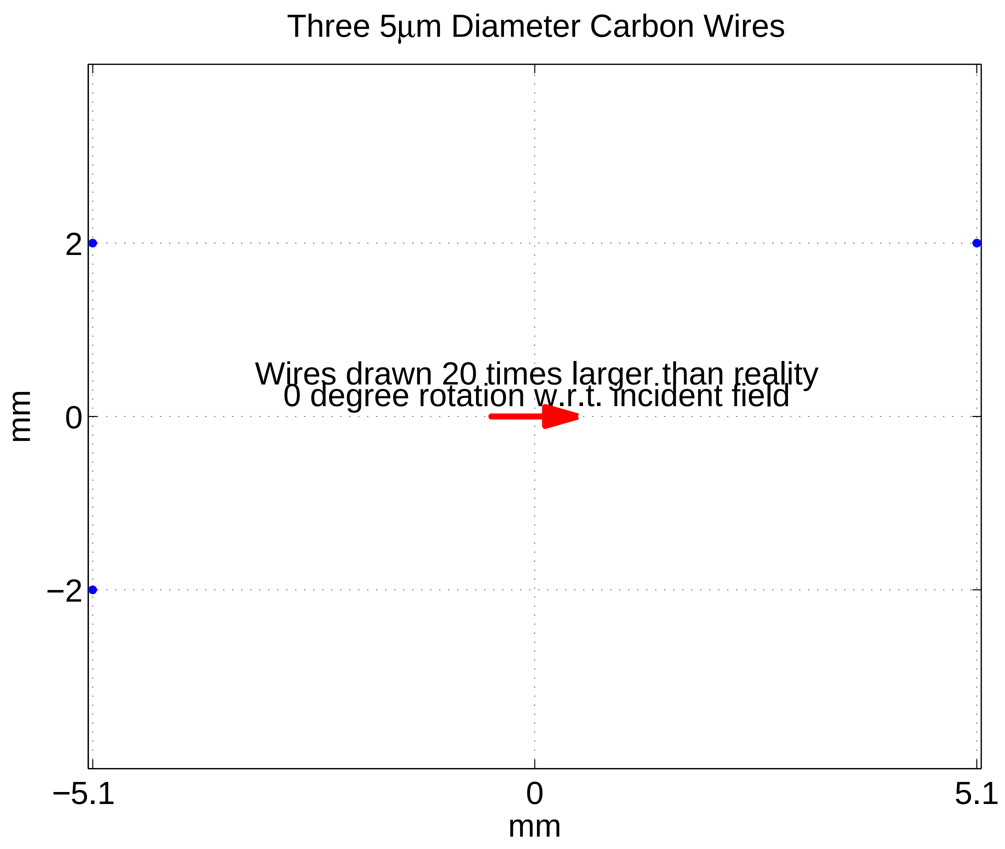


Figure 6: *Three carbon fiber arrangement.*

Radius (μm)	**Material**
337	Vacuum
372	0.833C+0.074H+0.043Br
1275	CH
1600	0.997Be+0.003Cu
1 μm **Radius Bump**	
Location (μm)	**Material**
(-1600,0)	0.833C+0.074H+0.043Br
(-1131.37,1131.37)	0.833C+0.074H+0.043Br
(0,1600)	0.833C+0.074H+0.043Br

Table 3: *Description of the multishell geometry and physical parameters. A figure of the multishell is presented in Figure 10.*

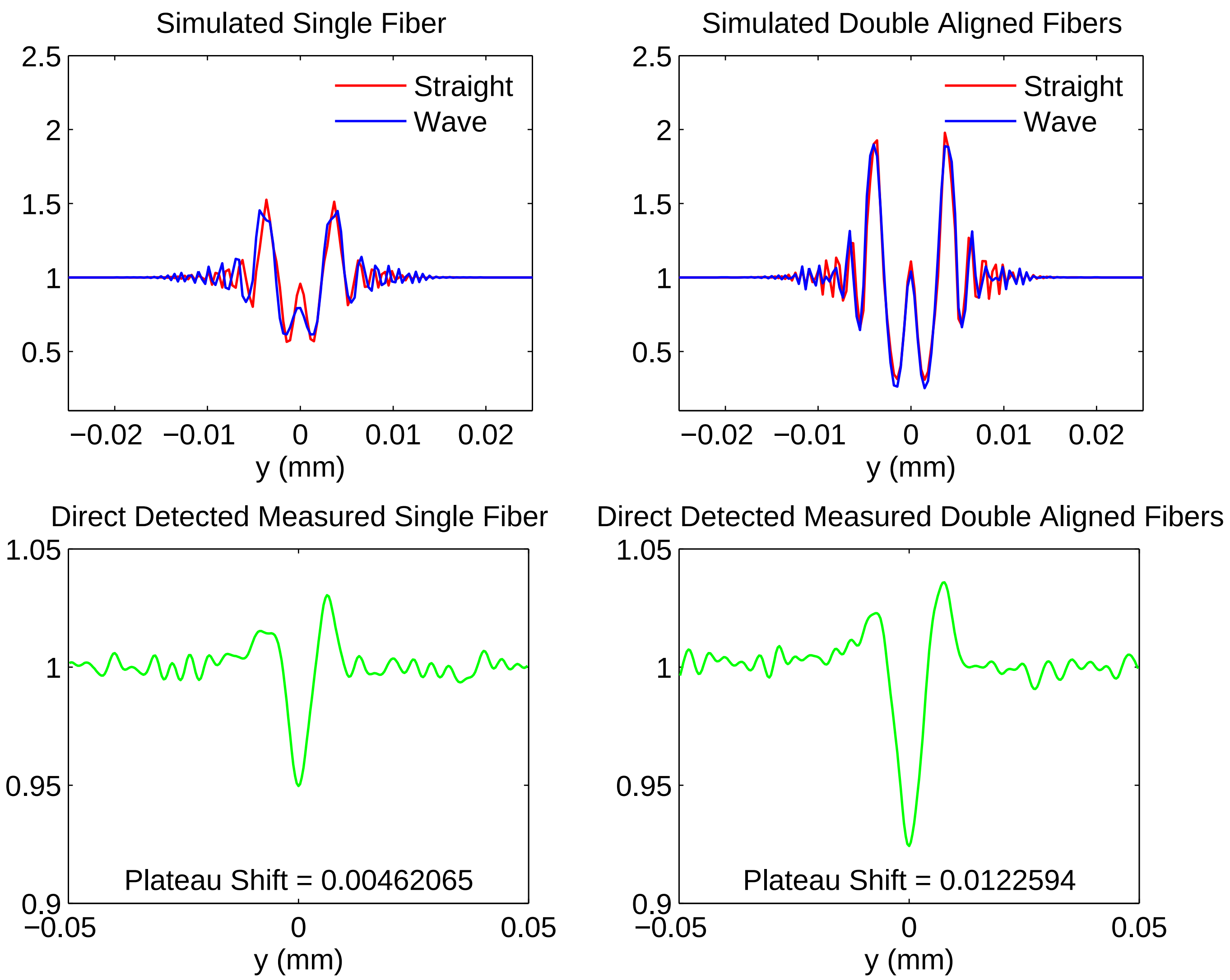
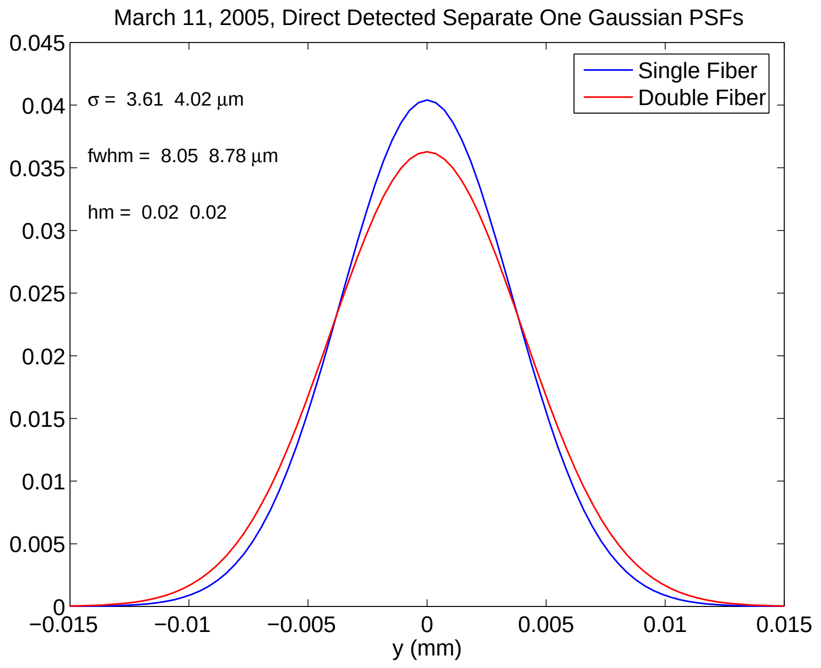
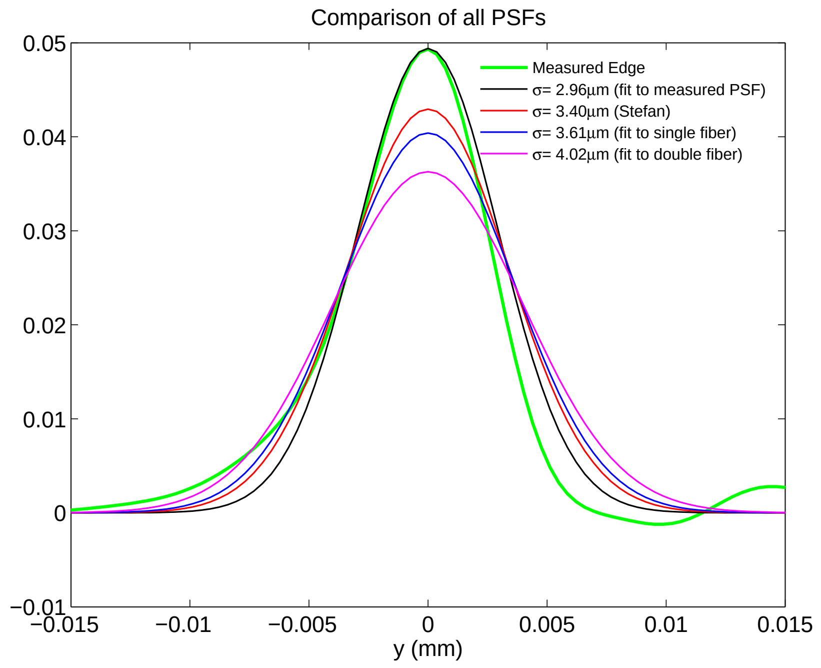


Figure 7: *Comparison between the full resolution simulated and measured March 11, 2005, 5 μm diameter carbon fiber data. The simulated data blurred with Au/Cu edge-based PSFs are presented in Figure 8. The same data blurred with PSFs* <u>*fit*</u> *to the data are shown in Figure 9.* **Note***: the measured data "plateau" was shifted up by the indicated value to center it about one.*

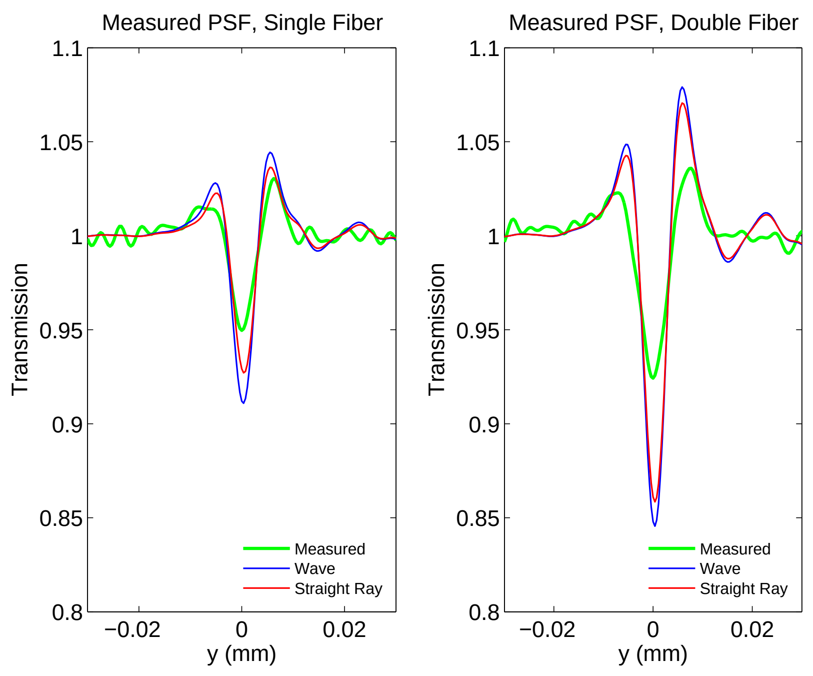


(a)

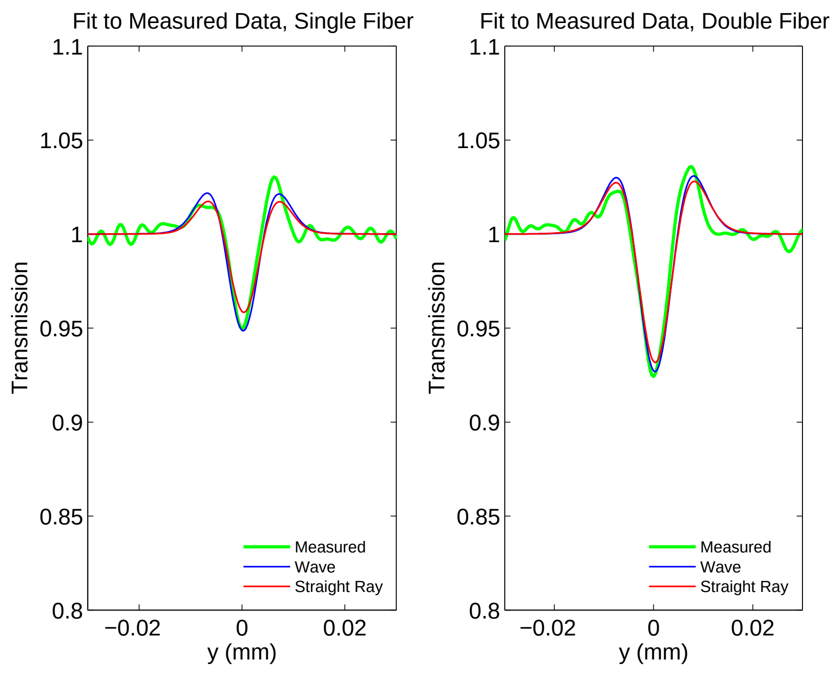


(b)

Figure 8: *Single Gaussian fits to the measured data sets. (a) fits to the single and double 5 μm diameter carbon fiber data. (b) comparison of all Gaussians.*



(a)



(b)

Figure 9: *(a) simulated 5μm diameter carbon fiber data convolved with the measured Au/Cu edge PSF plotted with the measured carbon fiber data. (b) simulated fiber data convolved with the Gaussians fit to the data. The Gaussians are presented in Figure 8(a).*

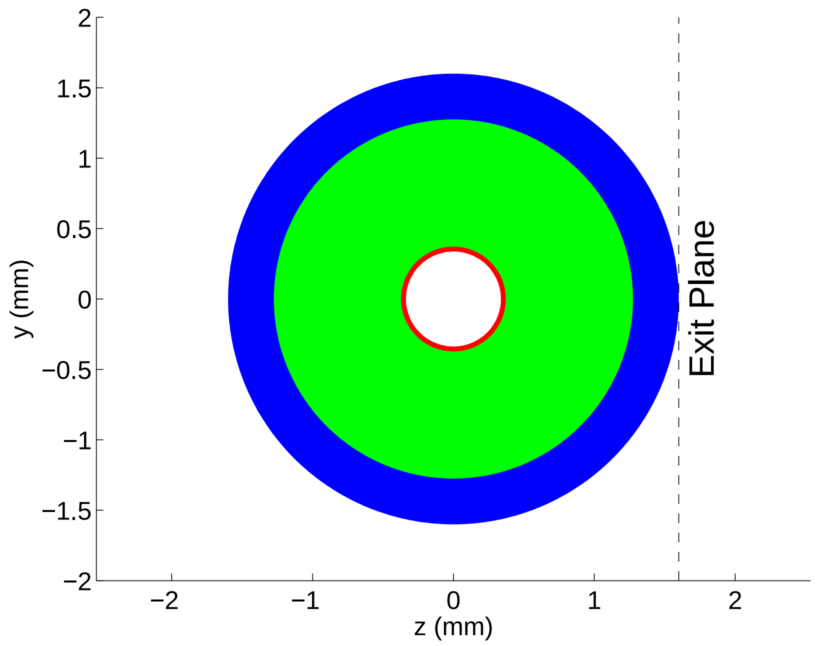


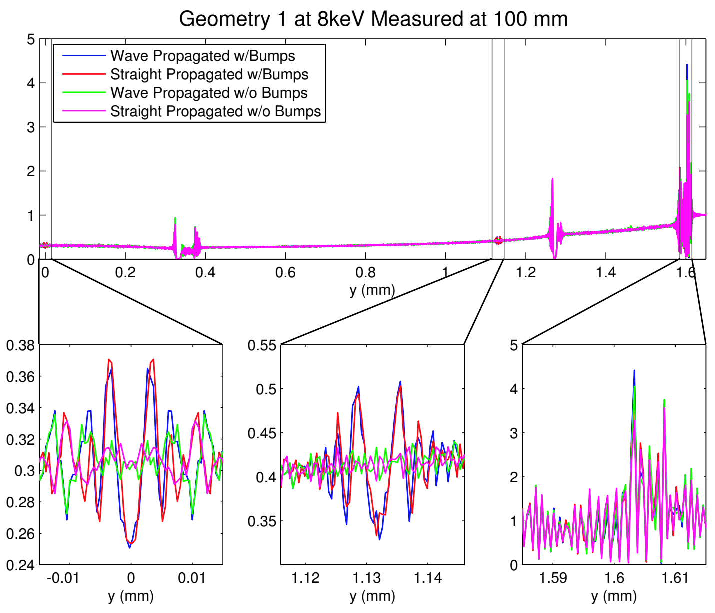
Figure 10: *Geometry of multishell.*

6 Conclusion

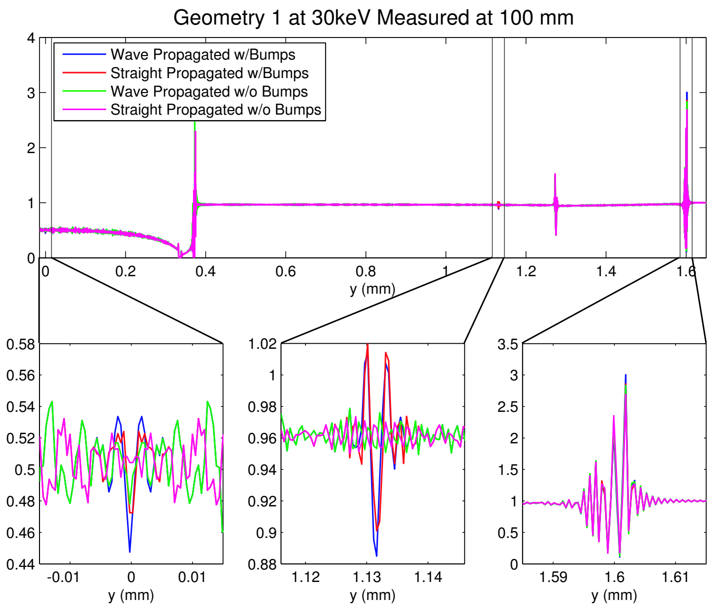
Given the simulated and limited experimental results we presented, we we believe diffraction is negligible, particularly so given the large detector blur. We note that the simulation code does not model density, only the refractive index parameters listed in Appendix B. Higher density materials substituted in the multishells or other configurations may result in observable diffraction effects.

7 Acknowledgments

We wish to thank Dr. David Chambers for suggesting the double obscuring copper edge configuration. We thank Dr. Jeffrey Kallman for the use of his codes.

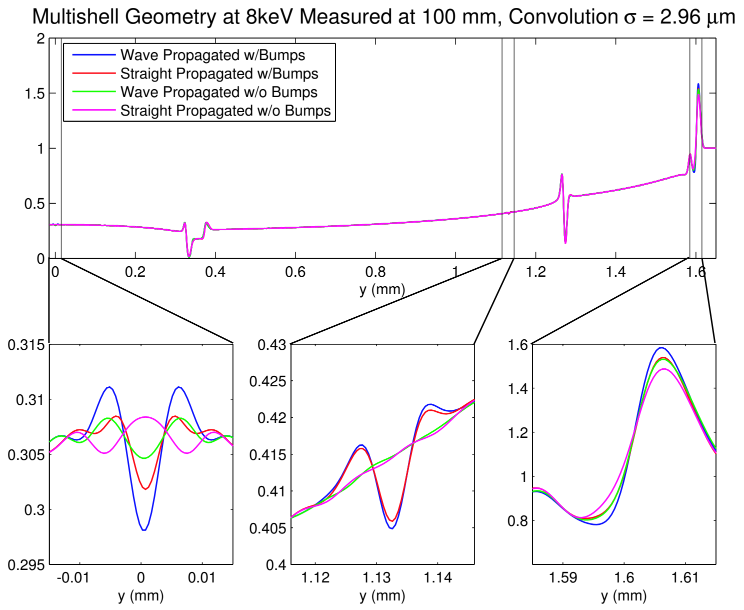


(a)

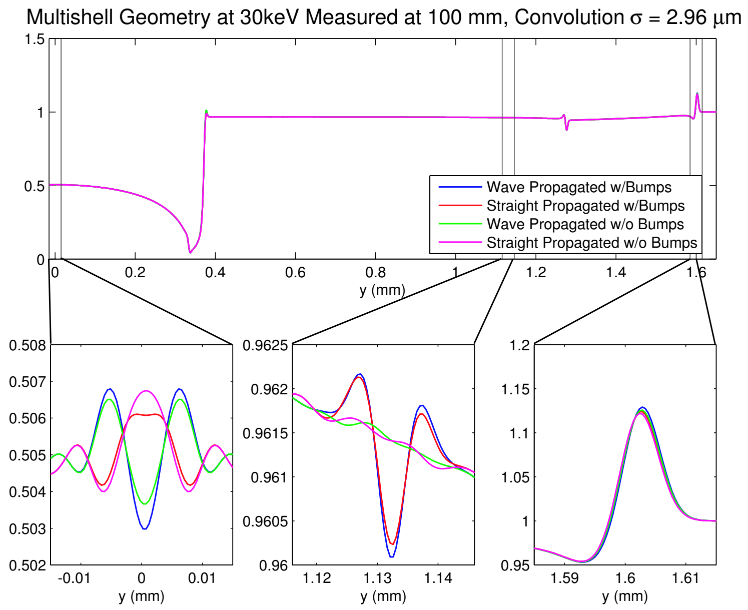


(b)

Figure 11: *Multishell results for (a) 8 keV and (b) 30 kev. The top plot of each graph set shows the top half of the received far field intensity. The three bottom plots show the intensity details around the area projected from the bumps. The far field intensity was evaluated at 100 mm.*



(a)



(b)

Figure 12: *Multishell results for (a) 8 keV and (b) 30 kev but with a 2.96 μm detector blur applied.*

A Double Gaussian PSF

We performed double Gaussian fits to the PSF using as a model,

$$h(y,\underline{\Omega}) = \left(\frac{1}{\sqrt{2\pi}\sum_{n=1}^{N_g}\alpha_n\sigma_n}\right)\sum_{n=1}^{N_g}\alpha_n\ \exp\left[-\frac{1}{2}\left(\frac{y-y_{0n}}{\sigma_n}\right)^2\right], \tag{15}$$

where N_g is the number of Gaussians (two), and the values for the scale (α_n), width (σ_n), and shift (y_{0n}) parameters form the parameter vector,

$$\underline{\Omega} \equiv \left[\{\alpha_n\}_{n=1}^{N_g} \quad \{\sigma_n\}_{n=1}^{N_g} \quad \{y_{0n}\}_{n=1}^{N_g}\right]^{\mathrm{T}}. \tag{16}$$

As described previously, the measured Au/Cu edge PSF was fit to the double Gaussian via

$$\hat{\underline{\Omega}} = \underset{\underline{\Omega}}{\text{argmin}}\sum_n\left[h_y(y_n)-h(y_n,\underline{\Omega})\right]^2, \tag{17}$$

and the carbon fiber data via,

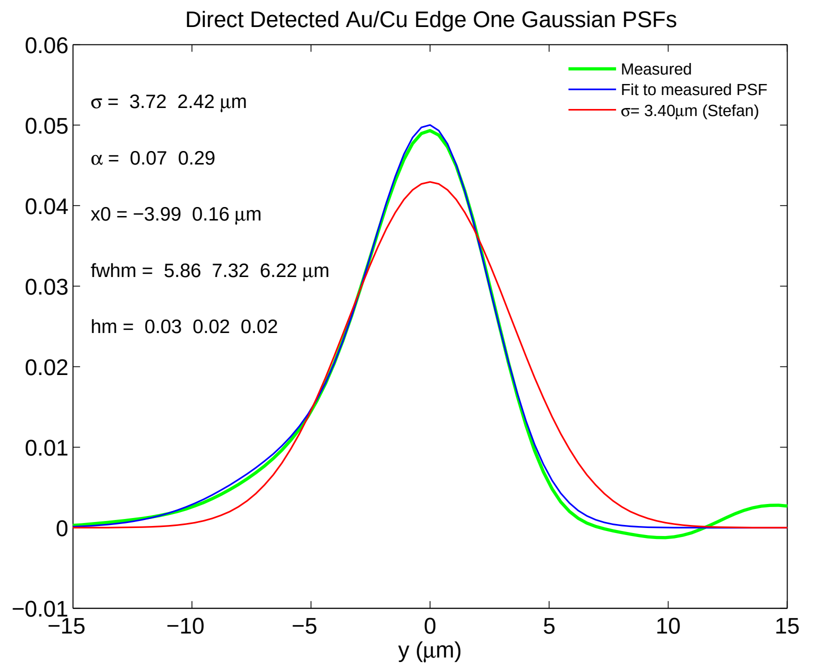
$$\hat{\underline{\Omega}} = \underset{\underline{\Omega}}{\text{argmin}}\sum_n\left[d(y_n)-\text{conv}(f(y_n),h(y_n,\underline{\Omega}))\right]^2. \tag{18}$$

The results are presented in Figures 13, 14, 15, and 16.

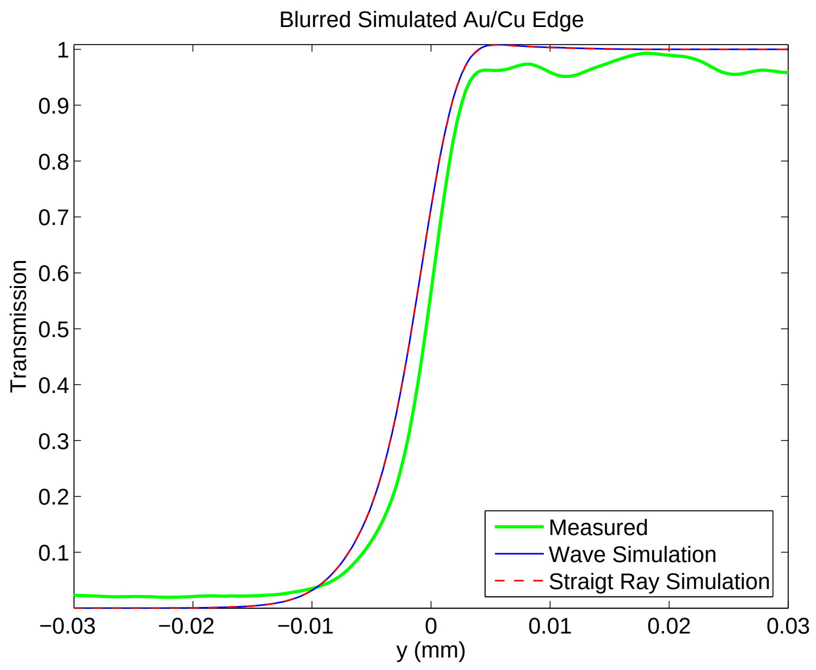
B Physical Parameter Lookup

Energy	**8 keV**	**30 keV**
λ	$1.55\times10^{-4}\mu$m	$4.14\times10^{-5}\mu$m

8 keV			
Material	δ	β	$1-\delta$
W	4.7008×10^{-5}	3.9664×10^{-6}	0.999952992
Be	5.3265×10^{-6}	2.0739×10^{-9}	0.9999946735
Cu	2.4759×10^{-5}	5.6196×10^{-7}	0.999975241
C ($\rho=1.8$ g/cc)	5.85×10^{-6}	9.46×10^{-9}	0.99999415
C ($\rho=2.2$ g/cc)	7.1526×10^{-6}	1.1560×10^{-8}	0.9999928474
H	5.7846×10^{-10}	6.5200×10^{-16}	0.99999999942154
Br	8.6972×10^{-6}	3.3997×10^{-7}	0.9999913028
Au	4.7730×10^{-5}	4.9592×10^{-6}	0.99995227
W	4.7008×10^{-5}	3.9664×10^{-5}	0.999952992
Fe	2.2676×10^{-5}	2.9621×10^{-6}	0.999977324
CH	3.5766×10^{-6}	5.7800×10^{-9}	0.99999642341077
0.997Be+0.003Cu	5.3848×10^{-6}	3.7536×10^{-9}	0.9999946152025
0.923C+0.077H	6.6019×10^{-6}	1.0670×10^{-8}	0.999993398105659
0.883C+0.074H+0.043Br	6.6898×10^{-6}	2.4826×10^{-8}	0.999993310231794
0.877C+0.073H+0.050Fe	7.4067×10^{-6}	1.5824×10^{-7}	0.999992593327572



(a)



(b)

Figure 13: *(a) double Gaussian fit to the measured Au/Cu edge PSF. (b) simulated data convolved with the double Gaussian.*

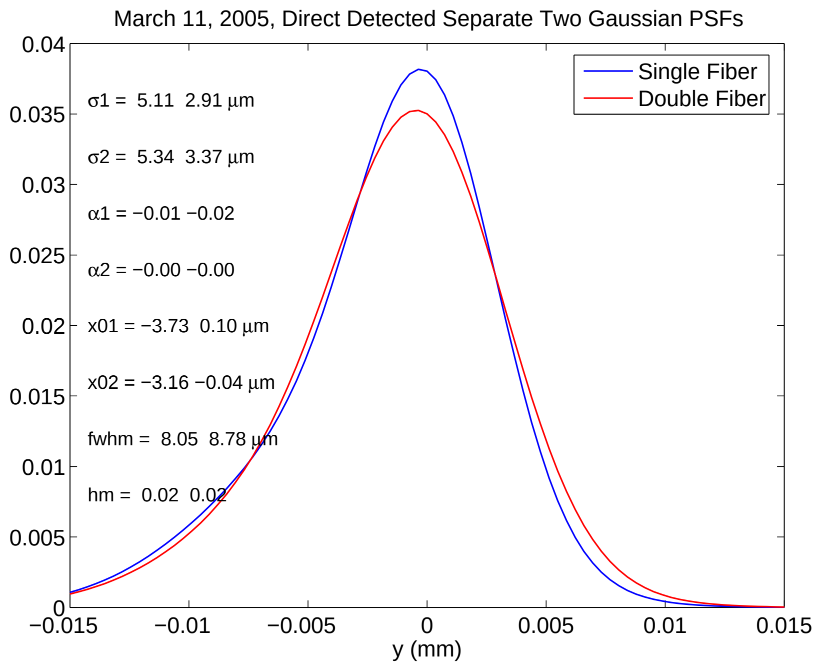


Figure 14: *Double Gaussian fits to the carbon fiber data.*

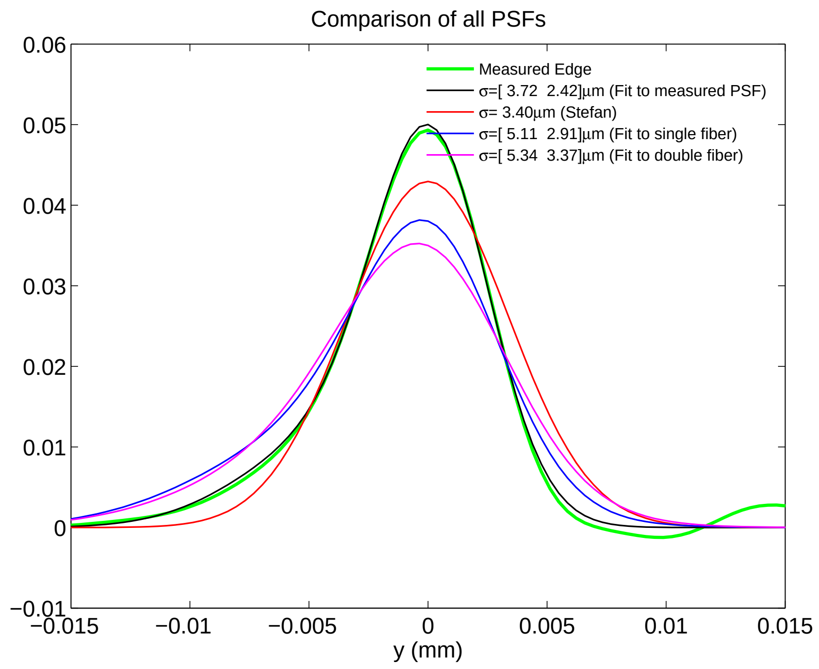
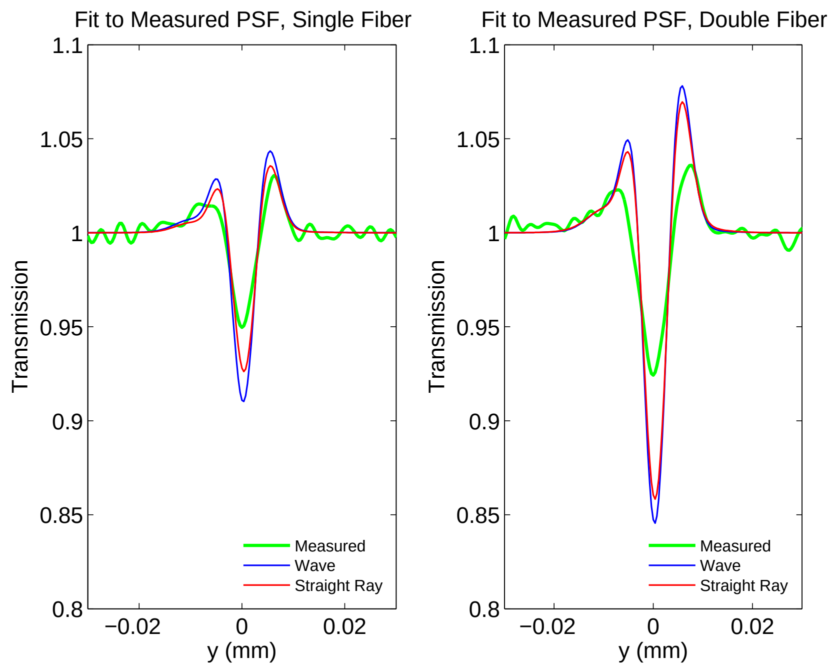
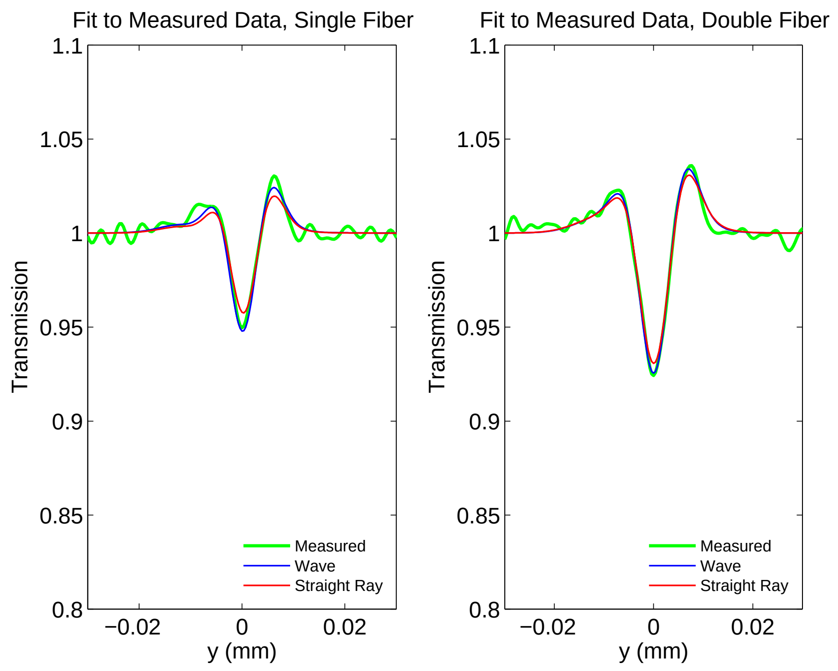


Figure 15: *Comparison of the double Gaussian PSFs.*



(a)



(b)

Figure 16: *Comparison of the measured carbon fiber data and the simulated data convolved with the double Gaussian PSFs. (a) double Gaussian fit to the measured edge data. (b) double Gaussian fit to the fiber data.*

30 keV			
Material	δ	β	$1-\delta$
W	3.5614×10^{-6}	1.3158×10^{-7}	0.9999964386
Be	3.7835×10^{-7}	6.9798×10^{-12}	0.99999962165
Cu	1.9030×10^{-6}	3.0749×10^{-8}	0.999998097
C	5.0699×10^{-7}	3.7394×10^{-11}	0.99999949301
H	4.1135×10^{-11}	1.8122×10^{-18}	0.999999999958865
Br	6.3585×10^{-7}	1.7888×10^{-8}	0.99999936415
Au	3.5555×10^{-6}	1.6406×10^{-7}	0.9999964445
W	3.5614×10^{-6}	1.3158×10^{-7}	0.9999964386
Fe	1.7041×10^{-6}	1.9762×10^{-8}	0.9999982959
CH	2.5352×10^{-7}	1.8697×10^{-11}	0.999999746484432
0.997Be+0.003Cu	3.8292×10^{-7}	9.9206×10^{-11}	0.99999961707605
0.923C+0.077H	4.6795×10^{-7}	3.4515×10^{-11}	0.999999532045063
0.883C+0.074H+0.043Br	4.7502×10^{-7}	8.0220×10^{-10}	0.999999524983236
0.877C+0.073H+0.050Fe	5.2984×10^{-7}	1.0209×10^{-9}	0.999999470161767

C MATLAB Codes

The following sections list the MATLAB codes used in the data processing and analysis.

C.1 Code to Read the Au/Cu Edge Data

```
%*****************************************************************************
%
% TITLE:    assemble.m
% AUTHOR:   Sean K. Lehman
% DATE:     June 27, 2005
% FUNCTION: Load and display the real & simulated x-ray data
% SYNTAX:
%
% MODIFICATIONS:
%
%
%    (c) Copyright 2005 the Regents of the University
%             of California. All rights reserved.
%
%     This work was produced at the Lawrence Livermore
%     National Laboratory. The United States Government
%     retains certain rights therein.
%
%***************************************************************************/
um    = 1e-6;
mm    = 1e-3;
Width = 0.1 * mm;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Load the simulated data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[yst683 dst683] = textread('683mm.s.int.dat','','headerlines',1);
[ywv683 dwv683] = textread('683mm.w.int.dat','','headerlines',1);
dys             = ( yst683(2) - yst683(1) ) * um;
dyw             = ( ywv683(2) - ywv683(1) ) * um;
yst683          = yst683*um ;
ywv683          = ywv683*um ;
if dys ~= dyw
  error(['The wave and straight-ray simulations have differing sample' ...
 ' intervals']);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% As a test, try resampling the simulated data from dy=0.366um to dy=0.5um
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if 0
  dy0     = dys;
  dy1     = 0.6*um;
  dy1     = 0.8*um;
  dy1     = 0.52*um;
  dy1     = 0.4*um;
  mag     = 1e4;
  p       = round( mag*dy0/um );
  q       = round( mag*dy1/um );
  dy      = (q/p)*dy0;
  fprintf(1,'Old Simulated dy        : %g micrometers\n',dy0/um);
  fprintf(1,'New Simulated dy        : %g micrometers\n',dy/um);
  fprintf(1,'[p q]                   : [%d %d]\n',p,q);
  dst     = resample( dst683-mean(dst683) , p , q ) + mean(dst683);
  dwv     = resample( dwv683-mean(dwv683) , p , q ) + mean(dwv683);
  Ny      = length(dwv);
  yst683  = [0:Ny-1]*dy;
  ywv683  = yst683;
  dst683  = dst;
  dwv683  = dwv;
  % Fix edge effects
  dst683(1:500) = 0;
  dst683(end-500:end) = 1;
  dwv683(1:500) = 0;
  dwv683(end-500:end) = 1;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Compute PSFs from the simulated data.
% These are used to center the data.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
psfwv683        = fd1( dwv683 , ywv683(2)-ywv683(1) );
psfwv683        = psfwv683 / sum(psfwv683);
psfst683        = fd1( dst683 , yst683(2)-yst683(1) );
psfst683        = psfst683 / sum(psfst683);

% Center the data about zero
[val,loc]       = max(abs(psfwv683));
ywv683          = ywv683 - ywv683(loc);
[val,loc]       = max(abs(psfst683));
yst683          = yst683 - yst683(loc);
nwv683          = find( abs(ywv683)<=Width/2 );
nst683          = find( abs(yst683)<=Width/2 );


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Save the simulated data.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Simulated683 = struct('y'  , ywv683(nwv683),...
      'wv' , dwv683(nwv683),...
      'st' , dst683(nst683),...
      'dy' , ywv683(2) - ywv683(1));

fprintf(1,'Simulated 683 [Ny dy]   : %d %g micrometers\n',...
length(Simulated683.y),Simulated683.dy/um);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Load the measured data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
file1       = 'lo_avg7_T_AuEdge-1157-6.5s-45kV-44ma-RIS.32bit.txt';
dyold       = 2.47*um;
[tmp y1 d1] = textread(file1,'%f %f %f');
y1          = y1*dyold;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Resample the measured data to the simulated data's interval
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(1,'Measured Data [Ny dy]   : %3d %g micrometers\n',...
length(y1),dyold/um);
mag   =   1000;
mag   =  10000;
p     = round( mag * dyold / um );
q     = round( mag * Simulated683.dy / um );
dy    = (q/p)*dyold;
fprintf(1,'Old Measured dy         : %g micrometers\n',dyold/um);
fprintf(1,'New Measured dy         : %g micrometers\n',dy/um);
fprintf(1,'[p q]                   : [%d %d]\n',p,q);
mu    = mean(d1);
dr    = resample(d1-mu,p,q) + mu;
Ny    = length(dr);
y     = [0:Ny-1]'*dy;y = y-mean(y);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Compute the PSF for both the pre- and post-resampled data
% Note: It is best to compute the PSF for the pre-resampled
% measured data and then resample that PSF than to compute the PSF
% from the post-resampled data.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
psfold      = fd1( d1 , dyold );
psf         = fd1( dr , dy );
% Resample the old psf
psfr        = resample(psfold-mean(psfold),p,q) + mean(psfold);
% Zero out edges where there are resampling problems
psf(1:200)       = 0;
psf(end-200:end) = 0;

psfr(1:200)       = 0;
psfr(end-200:end) = 0;

% Center the data about zero
[val,loc]    = max(abs(psfr));
y            = y - y(loc);
n            = find( abs(y)<=Width/2 );

Measured     = struct('y',y(n),'d',dr(n),'psf',psfr(n),'dy',y(2)-y(1));
Measured.psf = Measured.psf / sum( Measured.psf );

% Center the data about zero
[val,loc]   = max(abs(psfold));
y1          = y1 - y1(loc);
nold        = find( abs(y1)<=Width/2 );
psfold      = psfold(nold) / sum(psfold(nold));

psf         = psf(n) / sum(psf(n));

fprintf(1,'Simulated 683 Ny        : %g micrometers\n',...
length(Simulated683.y));
fprintf(1,'Measured Ny             : %d micrometers\n',...
length(Measured.y));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Compare via a plot the pre- and post-resampled data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fetchfigure('Pre- And Post-Resampled PSF 1');clf
hp = plot(y1(nold)/um,psfold/dyold,'g',...
  Measured.y/um,Measured.psf/dy,'b',...
  Measured.y/um,psf/dy,'r');
ht = [ title('Direct Detected Edge PSF'); xlabel('y (\mum)'); ylabel('Transmission') ];
set(gca,'xlim',Width*[-1 1]/4/um);
set(ht,'fontsize',14)
set(gca,'fontsize',14)
set(hp,'linewidth',1)
legend('PSF from data',...
       'Resampled PSF from data',...
       'PSF from resampled data');

fetchfigure('Pre- And Post-Resampled PSF');clf
hp = plot(y1(nold)/um,psfold/dyold,'g',...
  Measured.y/um,Measured.psf/dy);
ht = [ title('Direct Detected Edge PSF'); xlabel('y (\mum)'); ylabel('Transmission') ];
set(gca,'xlim',Width*[-1 1]/4/um);
set(ht,'fontsize',14)
set(gca,'fontsize',14)
set(hp,'linewidth',1);set(hp(1),'linewidth',2);
legend('PSF from data',...
       'Resampled PSF from data');
legend boxoff
```

```
cmd = 'print -depsc LANLAuCuEdgeResample.eps';
fprintf(1,'%s\n',cmd); eval( cmd );

save LANL Measured Simulated683

y = Measured.y;

fetchfigure('2005.06.23.LANL');clf
   hp = plot(y/mm,Measured.d,'g',...
     y/mm,Simulated683.wv,'b',...
     y/mm,Simulated683.st,'r--');
   ht = [ title('Direct Detected Edge Data'); xlabel('y (mm)'); ylabel('Transmission') ];
   set(gca,'xlim',Width*[-1 1]/4/mm);
   set(ht,'fontsize',14)
   set(gca,'fontsize',14)
   set(hp,'linewidth',1);set(hp(1),'linewidth',2)
   legend('Measured',...
  '683 Wave Simulation','683 Straight-Ray Simulation',...
  'location','SouthEast')
   legend boxoff

   drawnow; refresh

cmd = 'print -dpsc LANLAuCuEdge.ps';
cmd = 'print -depsc LANLAuCuEdge.eps';
fprintf(1,'%s\n',cmd); eval( cmd );
```

C.2 Code to Read the Carbon Fiber Data

```
%******************************************************************************
%
% TITLE:    assembleLANL.m
% AUTHOR:   Sean K. Lehman
% DATE:     August 24, 2005
% FUNCTION: Assemble measured and simulated data
%
%
% SYNTAX:
%
% MODIFICATIONS:
%
%
%    (c) Copyright 2004 the Regents of the University
%             of California. All rights reserved.
%
%     This work was produced at the Lawrence Livermore
%     National Laboratory. The United States Government
%     retains certain rights therein.
%
%*****************************************************************************/
mm     = 1e-3;
um     = 1e-6;
nm     = 1e-9;
Width  = 0.1*mm;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Load the simulated data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[ys ds] = textread('5um.st.int.dat','','headerlines',1);
[yw dw] = textread('5um.wv.int.dat','','headerlines',1);
ys      = ys * um; dys = ys(2) - ys(1);
yw      = yw * um; dyw = yw(2) - yw(1);
if dys ~= dyw
  error(['The wave and straight-ray simulations have differing sample' ...
 ' intervals']);
```

```
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Separate the one wire response from the two wire response.
% The single wire is at y==-2 mm.
% The double wires are at y==2 mm.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
yc     = [-2 2]*mm;
yrange = [ yc(1)-Width/2 yc(1)+Width/2
   yc(2)-Width/2 yc(2)+Width/2
 ];
n1     = find( (yw>=yrange(1,1)) & (yw<=yrange(1,2)) );
n2     = find( (yw>=yrange(2,1)) & (yw<=yrange(2,2)) );

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Resample the fiber data to the Au/Cu sample interval of 0.366
% micrometers.
% The simlated fiber data have a sample interval of 0.183 micrometers.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
mag    = 100000;
mag    =  10000;
load ../../AuCuEdge/2005.06.23.LANL/LANL Simulated683
dyAuCu = Simulated683.dy;
p   = fix( mag * dys / um );
q   = fix( mag * dyAuCu  / um );
dy  = (q/p)*dys;
fprintf(1,’Old simulated dy        : %g micrometers\n’,dys/um);
fprintf(1,’New simulated dy        : %g micrometers\n’,dy/um);
fprintf(1,’[p q]                   : [%d %d]\n’,p,q);
dsr1  = resample(ds(n1)-mean(ds(n1’)),p,q) + mean(ds(n1));
dsr2  = resample(ds(n2)-mean(ds(n2)),p,q) + mean(ds(n2));
dwr1  = resample(dw(n1)-mean(dw(n1)),p,q) + mean(dw(n1));
dwr2  = resample(dw(n2)-mean(dw(n2)),p,q) + mean(dw(n2));
Ny    = length(dsr1);
y     = [0:Ny-1]’*dy;y = y-mean(y);



% Extract the data range and center it
SimSingle = struct(’name’,’3/11/2005 LANL, 0 rotation, single wire’,...
   ’y’   , y,...
   ’st’  , dsr1,...
   ’wv’  , dwr1,...
   ’dy’  , dy);
SimSingle.y = SimSingle.y - mean( SimSingle.y );

SimDouble = struct(’name’,’3/11/2005 LANL, 0 rotation, double wire’,...
   ’y’   , y,...
   ’st’  , dsr2,...
   ’wv’  , dwr2,...
   ’dy’  , dy);
SimDouble.y = SimDouble.y - mean( SimDouble.y );

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Load the measured data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[tmp y d] = textread(’lo1_avg73_T_CarbonWire-031105+220.32bit.txt’,’%f %f %f’);
yold      = y * mm;
dyold     = yold(2) - yold(1);
Nyold     = length(d);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Resample the measured data to the simulated data’s interval
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(1,’Single Simulated [Ny dy]: %3d %g micrometers\n’,...
length(SimSingle.y),SimSingle.dy/um);
```

```
fprintf(1,'Double Simulated [Ny dy]: %3d %g micrometers\n',...
length(SimDouble.y),SimDouble.dy/um);
fprintf(1,'Measured Data [Ny dy]   : %3d %g micrometers\n',length(y),dyold/um);
mag = 1000;
p   = fix( mag * dyold / um );
q   = fix( mag * SimDouble.dy / um );
dy  = (q/p)*dyold;
fprintf(1,'New Measured dy         : %g micrometers\n',dy/um);
fprintf(1,'[p q]                   : [%d %d]\n',p,q);
mu    = mean(d);
dr    = resample(d-mu,p,q) + mu;
Ny    = length(dr);
y     = [0:Ny-1]'*dy;y = y-mean(y);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Extract and center the data single and double fiber responses
% Double is 1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[val l1]  = min( dr(1:fix(Ny/2)) );
n         = l1 + [-fix(Width/dy/2):fix(Width/dy/2)];
d1        = dr( n );
y1        = y( n );
y1        = y1 - mean( y1 );
dy1       = y1(2) - y1(1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Single is 2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[val l2]  = min( dr(fix(Ny/2):Ny) ); l2 = l2 + fix(Ny/2) - 1;
n         = l2 + [-fix(Width/dy/2):fix(Width/dy/2)];
d2        = dr( n );
y2        = y( n );
y2        = y2 - mean( y2 );
dy2       = y2(2) - y2(1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Detrend the data to place the plateau at one
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
p1 = polyfit(y1/mm,d1,0);
d1 = d1 - p1 + 1;

p2 = polyfit(y2/mm,d2,0);
d2 = d2 - p2 + 1;

MeasuredSingle = struct('y', y2, 'd', d2, 'dy', dy);
MeasuredDouble = struct('y', y1, 'd', d1, 'dy', dy);

save LANL.03.11.2005.mat SimSingle SimDouble MeasuredSingle MeasuredDouble
clear y1 y2 d1 d2 dr
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Repeat the extraction for the pre-resampled data
% Extract and center the data single and double fiber responses
% Double is 1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[val l1]  = min( d(1:fix(Nyold/2)) );
n         = l1 + [-fix(Width/dyold/2):fix(Width/dyold/2)];
d1        = d( n );
y1        = yold( n );
y1        = y1 - mean( y1 );
dyold1    = y1(2) - y1(1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Single is 2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[val l2]  = min( d(fix(Nyold/2):Nyold) ); l2 = l2 + fix(Nyold/2) - 1;
n         = l2 + [-fix(Width/dyold/2):fix(Width/dyold/2)];
d2        = d( n );
y2        = yold( n );
```

```
y2        = y2 - mean( y2 );
dyold2    = y2(2) - y2(1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Detrend the data to place the plateau at one
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
p1 = polyfit(y1/mm,d1,0);
d1 = d1 - p1 + 1;

p2 = polyfit(y2/mm,d2,0);
d2 = d2 - p2 + 1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Compare via a plot the pre- and post-resampled data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fetchfigure( 'Compare Pre- & Post-Resample' );clf
ylim = [0 2.5];

subplot(121)
   hp = plot(MeasuredSingle.y/mm,MeasuredSingle.d,'b',...
     y2/mm,d2,'r');
   ht = [ xlabel('y (mm)');
  title('Single Fiber') ];
   set(hp,'linewidth',1);set(gca,'fontsize',12);set(ht,'fontsize',12);
   set(gca,'ylim',[0.9 1.05]);
   htxt = text(0,0.91,sprintf('Plateau Shift = %g',p2));
   set(htxt,'fontsize',12,'horizontal','center')
   legend('Post Resample','Pre Resample');
   legend boxoff

subplot(122)
   hp = plot(MeasuredDouble.y/mm,MeasuredDouble.d,'b',...
     y1/mm,d1,'r');
   ht = [ xlabel('y (mm)');
  title('Double Aligned Fibers') ];
   set(hp,'linewidth',1);set(gca,'fontsize',12);set(ht,'fontsize',12);
   set(gca,'ylim',[0.9 1.05]);
   htxt = text(0,0.91,sprintf('Plateau Shift = %g',p1));
   set(htxt,'fontsize',12,'horizontal','center')
   legend('Post Resample','Pre Resample');
   legend boxoff

cmd = 'print -depsc LANL.03.11.2005.Resample.eps';
fprintf(1,'%s\n',cmd); eval( cmd );
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fetchfigure( '03/11/2005 LANL Three Carbon Fibers' );clf

ylim = [0 2.5];

subplot(221)
   hp = plot(SimSingle.y/mm,SimSingle.st,'r',...
     SimSingle.y/mm,SimSingle.wv,'b');
   ht = [ xlabel('y (mm)');
  title('Simulated Single Fiber') ];
   set(hp,'linewidth',1);set(gca,'fontsize',12);set(ht,'fontsize',12);
   set(gca,'ylim',[0.1 2.5]);
   set(gca,'xlim',Width*[-1 1]/mm/4);
   legend('Straight','Wave');
   legend boxoff

subplot(222)
   hp = plot(SimDouble.y/mm,SimDouble.st,'r',...
     SimDouble.y/mm,SimDouble.wv,'b');
   ht = [ xlabel('y (mm)');
  title('Simulated Double Aligned Fibers') ];
   set(hp,'linewidth',1);set(gca,'fontsize',12);set(ht,'fontsize',12);
   set(gca,'xlim',Width*[-1 1]/mm/4);
   set(gca,'ylim',[0.1 2.5]);
   legend('Straight','Wave');
   legend boxoff

subplot(223)
   hp = plot(MeasuredSingle.y/mm,MeasuredSingle.d,'g');
   ht = [ xlabel('y (mm)');
  title('Direct Detected Measured Single Fiber') ];
   set(hp,'linewidth',1);set(gca,'fontsize',12);set(ht,'fontsize',12);
   set(gca,'ylim',[0.9 1.05]);
   htxt = text(0,0.91,sprintf('Plateau Shift = %g',p2));
   set(htxt,'fontsize',12,'horizontal','center')

subplot(224)
   hp = plot(MeasuredDouble.y/mm,MeasuredDouble.d,'g');
   ht = [ xlabel('y (mm)');
  title('Direct Detected Measured Double Aligned Fibers') ];
   set(hp,'linewidth',1);set(gca,'fontsize',12);set(ht,'fontsize',12);
   set(gca,'ylim',[0.9 1.05]);
   htxt = text(0,0.91,sprintf('Plateau Shift = %g',p1));
   set(htxt,'fontsize',12,'horizontal','center')


cmd = 'print -depsc LANL.03.11.2005.eps';
fprintf(1,'%s\n',cmd); eval( cmd );
```

C.3 Code to Process the Data Sets I

This code, `Main1.m`, fits the data to single Gaussians.

```
%****************************************************************************
%
% TITLE:    Main1.m
% AUTHOR:   Sean K. Lehman
% DATE:     September 20, 2005
% FUNCTION: Process the data from the
%               AuCuEdge/2005.06.23.LANL
%           and
%               ThreeCarbonWires/2005.03.11.LANL
%           directories
%           Fit single Gaussian PSFs.
% SYNTAX:
% CALLS:
%
% MODIFICATIONS:
%
%
%    (c) Copyright 2005 the Regents of the University
%              of California. All rights reserved.
%
%     This work was produced at the Lawrence Livermore
%     National Laboratory. The United States Government
%     retains certain rights therein.
%
%***************************************************************************/
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% AuCuEdge/2005.06.23.LANL/LANL.mat contains
% Measured =
%      y: [273x1 double]
%      d: [273x1 double]
%    psf: [273x1 double]
%     dy: 3.6600e-07
```

```
%
% Simulated683 =
%      y: [273x1 double]
%     wv: [273x1 double]
%     st: [273x1 double]
%     dy: 3.6600e-07
%
% Simulated716 =
%      y: [273x1 double]
%     wv: [273x1 double]
%     st: [273x1 double]
%     dy: 3.6600e-07
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
load ../AuCuEdge/2005.06.23.LANL/LANL.mat
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ThreeCarbonWires/2005.03.11.LANL/LANL.03.11.2005.mat contains
% MeasuredDouble =
%      y: [273x1 double]
%      d: [273x1 double]
%     dy: 3.6700e-07
%
% MeasuredSingle =
%      y: [273x1 double]
%      d: [273x1 double]
%     dy: 3.6700e-07
%
% SimDouble =
%     name: ’3/11/2005 LANL, 0 rotation, double wire’
%        y: [273x1 double]
%       st: [273x1 double]
%       wv: [273x1 double]
%       dy: 3.6700e-07
%
% SimSingle =
%     name: ’3/11/2005 LANL, 0 rotation, single wire’
%        y: [273x1 double]
%       st: [273x1 double]
%       wv: [273x1 double]
%       dy: 3.6700e-07
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
load ../ThreeCarbonWires/2005.03.11.LANL/LANL.03.11.2005.mat
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Preliminary definitions
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
um    = 1e-6;
mm    = 1e-3;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Range to be plotted
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Width = 0.06 * mm;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Fit the data using single Gaussians
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
sigma0 = 2.5 * um;
sigma0 = 3 * um;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Tinker with options
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
opts   = optimset(optimset(’fminsearch’),...
  ’tolx’,1e-10,...
  ’tolfun’,1e-10,...
  ’MaxIter’,    400*length(sigma0),...
  ’MaxFunEvals’,600*length(sigma0));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Solve for the Gaussian fit
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
y    = Measured.y;
md   = Measured.d;
psfm = Measured.psf;
sd   = Simulated683.wv;
dy   = Measured.dy;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Sigma1 is fit by blurring the edge data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[sigma1,fval,iflag,output]=fminsearch(@(o)FitLANLEdge(o,y,md,sd),sigma0,opts);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Sigma2 is fit directly to the PSF. There is a difference.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[sigma2,fval,iflag,output]=fminsearch(@(o)FitLANLPSF(o,y,psfm),sigma0,opts);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Define Stefan’s sigma
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
sigmas = 3.4 * um;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Print results
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(1,’fval  : %g\n’,fval);
fprintf(1,’iflag : %d\n’,iflag);
fprintf(1,’Output Message: %s\n’,output.message);
fprintf(1,’sigma = ’);fprintf(1,’%g ’,[sigma1 sigma2]/um);fprintf(1,’micrometers\n’);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Compute the PSFs
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
psf1  = multigaussian(y,[],sigma1,0);
psf1  = psf1/sum(psf1);
psf2  = multigaussian(y,[],sigma2,0);
psf2  = psf2/sum(psf2);
psfs  = multigaussian(y,[],sigmas,0);
psfs  = psfs/sum(psfs);

fprintf(1,’[sum(psf1) max(psf1) dy/(sqrt(2pi sigma1)] = [%g %g %g]\n’,...
sum(psf1),max(psf1),dy/(sqrt(2*pi)*sigma1));
fprintf(1,’[sum(psf2) max(psf2) dy/(sqrt(2pi sigma2)] = [%g %g %g]\n’,...
sum(psf2),max(psf2),dy/(sqrt(2*pi)*sigma2));
fprintf(1,’[sum(psfs) max(psfs) dy/(sqrt(2pi sigmas)] = [%g %g %g]\n’,...
sum(psfs),max(psfs),dy/(sqrt(2*pi)*sigmas));

[fwhm1 hm1]  = FullWidthHalfMax(y/mm,psf1);
[fwhm2 hm2]  = FullWidthHalfMax(y/mm,psf2);
[fwhm3 hm3]  = FullWidthHalfMax(y/mm,psfs);
[fwhm4 hm4]  = FullWidthHalfMax(y/mm,Measured.psf);
fwhm         = [diff(fwhm1)*mm/um
diff(fwhm2)*mm/um
diff(fwhm3)*mm/um
diff(fwhm4)*mm/um
       ];
hm           = [ hm1 hm2 hm3 hm4 ];
pstr = { [sprintf(’\\sigma = ’) sprintf(’%5.2f ’,[sigma1 sigma2]/um) sprintf(’\\mum’)];
 [sprintf(’fwhm = ’) sprintf(’%5.2f ’,fwhm) sprintf(’\\mum’)];
 [sprintf(’hm = ’) sprintf(’%5.2f ’,hm)];
       };

str = ’Direct Detected Au/Cu Edge One Gaussian PSFs’;
cmd = sprintf(’fetchfigure(’’%s’’);clf’,str);
fprintf(1,’%s\n’,cmd); eval( cmd );

hp = plot(y/mm,psfm,’g’,...
  y/mm,psf1,’b’,...
  y/mm,psf2,'k',...
  y/mm,psfs,'r');
set(gca,'xlim',Width*[-1 1]/4/mm)
set(hp,'linewidth',1);set(hp(1),'linewidth',2)
ht = [ title(str); xlabel('y (mm)') ];
set(ht,'fontsize',14)
set(gca,'fontsize',14)

xlim  = get(gca,'xlim');  Dxlim = diff(xlim);
ylim  = get(gca,'ylim');  Dylim = diff(ylim);
dytxt = 0.1 * Dylim;
for m=1:length(pstr)
  htxt = text(xlim(1)+0.025*Dxlim,ylim(2)-m*dytxt,pstr(m));
  set(htxt,'fontsize',12);
end
hleg = legend('Measured',...
      sprintf('\\sigma=%5.2f\\mum (fit to blurred data)',sigma1/um),...
      sprintf('\\sigma=%5.2f\\mum (fit to measured PSF)',sigma2/um),...
      sprintf('\\sigma=%5.2f\\mum (Stefan)',sigmas/um));
set(hleg,'box','off','fontsize',10)
cmd = 'print -depsc Figure01.eps';
fprintf(1,'%s\n',cmd); eval(cmd);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Blur the simulated edge data with the sigma2 Gaussian fit
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
omega2           = [sum(multigaussian(y,[],sigma2,0)) sigma2 0];
Simulated683.wvb = ApplyMultiGaussianBlur(y,Simulated683.wv,omega2);
Simulated683.stb = ApplyMultiGaussianBlur(y,Simulated683.st,omega2);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Plot edge blurred data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fetchfigure('LANL Au/Cu Blurred');clf
   hp = plot(y/mm,Measured.d,'g',...
     y/mm,Simulated683.wvb,'b',...
     y/mm,Simulated683.stb,'r--');
   set(hp,'linewidth',1);
   set(hp(1),'linewidth',2);
   axis tight
   set(gca,'xlim',Width*[-1 1]/2/mm)
   ht = [ xlabel('y (mm)');
  ylabel('Transmission');
  title('Blurred Simulated Au/Cu Edge')];
   set(ht,'fontsize',14)
   set(gca,'fontsize',14)
   hleg = legend('Measured',...
 'Wave Simulation',...
 'Straight Ray Simulation','location','SouthEast');

cmd = 'print -depsc Figure02.eps';
fprintf(1,'%s\n',cmd); eval(cmd);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Now start working with the carbon fiber data.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(1,'Measured single dy : %g micrometers\n',MeasuredSingle.dy/um);
fprintf(1,'Measured double dy : %g micrometers\n',MeasuredDouble.dy/um);
fprintf(1,'Simulated single dy: %g micrometers\n',SimSingle.dy/um);
fprintf(1,'Simulated double dy: %g micrometers\n',SimDouble.dy/um);

yf   = MeasuredDouble.y;
dyf  = MeasuredDouble.dy;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Measured data single wire
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
mds = MeasuredSingle.d;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Measured data double wire
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
mdd = MeasuredDouble.d;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Simulated data single wire
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
sds = SimSingle.wv;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Simulated data double wire
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
sdd = SimDouble.wv;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Fit the data.
% Fit the single and double wires separately
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Values for initial guess
sigma0     = 3 * [ 1 1 ] * um;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Tinker with options
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
opts       = optimset(optimset('fminsearch'),...
      'tolx',1e-10,...
      'tolfun',1e-10,...
      'MaxIter',    400*length(sigma0),...
      'MaxFunEvals',600*length(sigma0));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Solve for the Gaussian fit
[sigmaf,fval,iflag,output]=fminsearch(@(o)FitLANLFiber(o,yf,mds,mdd,sds,sdd),...
      sigma0,opts);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(1,'fval  : %g\n',fval);
fprintf(1,'iflag : %d\n',iflag);
fprintf(1,'Output Message: %s\n',output.message);
fprintf(1,'sigma = ');fprintf(1,'%g ',sigmaf/um);fprintf(1,'micrometers\n');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Compute the fitted PSF
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
psff1 = multigaussian(yf,[],sigmaf(1),0);
psff2 = multigaussian(yf,[],sigmaf(2),0);
fprintf(1,'[sum(psff1) max(psff1) dy/(sqrt(2pi sigma1)] = [%g %g %g]\n',...
sum(psff1),max(psff1),dyf/(sqrt(2*pi)*sigmaf(1)));
fprintf(1,'[sum(psff2) max(psff2) dy/(sqrt(2pi sigma2)] = [%g %g %g]\n',...
sum(psff2),max(psff2),dyf/(sqrt(2*pi)*sigmaf(2)));
[fwhm1 hm1] = FullWidthHalfMax(yf/mm,psff1);
[fwhm2 hm2] = FullWidthHalfMax(yf/mm,psff2);
fwhm        = [diff(fwhm1)*mm/um diff(fwhm2)*mm/um];
hm          = [ hm1 hm2 ];
pstr = { [sprintf('\\sigma = ') sprintf('%5.2f ',sigmaf/um) sprintf('\\mum')];
 [sprintf('fwhm = ') sprintf('%5.2f ',fwhm) sprintf('\\mum')];
 [sprintf('hm = ') sprintf('%5.2f ',hm)];
       };

str = 'March 11, 2005, LANL Separate One Gaussian PSFs';
cmd = sprintf('fetchfigure(''%s'');clf',str);
fprintf(1,'%s\n',cmd); eval( cmd );

hp = plot(yf/mm,psff1,'b',yf/mm,psff2,'r');
set(gca,'xlim',Width*[-1 1]/4/mm)
set(hp,'linewidth',1)
ht = [ title(str); xlabel('y (mm)') ];
```

```
set(ht,'fontsize',14)
set(gca,'fontsize',14)

xlim  = get(gca,'xlim');  Dxlim = diff(xlim);
ylim  = get(gca,'ylim');  Dylim = diff(ylim);
dytxt = 0.1 * Dylim;
for m=1:length(pstr)
  htxt = text(xlim(1)+0.025*Dxlim,ylim(2)-m*dytxt,pstr(m));
  set(htxt,'fontsize',12);
end
legend('Single Fiber','Double Fiber');
cmd = 'print -depsc Figure03.eps';
fprintf(1,'%s\n',cmd); eval(cmd);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Plot all the PSFs together
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fetchfigure('All PSFs');clf
hp = plot(y/mm , psfm,'g',...
  y/mm , psf2,'k',...
  y/mm , psfs,'r',...
  yf/mm, psff1,'b',...
  yf/mm, psff2,'m');
set(gca,'xlim',Width*[-1 1]/4/mm)
set(hp,'linewidth',1);set(hp(1),'linewidth',2)
ht = [ title('Comparison of all PSFs'); xlabel('y (mm)') ];
set(ht,'fontsize',14)
set(gca,'fontsize',14)

hleg = legend('Measured Edge',...
      sprintf('\\sigma=%5.2f\\mum (fit to measured PSF)',sigma2/um),...
      sprintf('\\sigma=%5.2f\\mum (Stefan)',sigmas/um),...
      sprintf('\\sigma=%5.2f\\mum (fit to single fiber)',sigmaf(1)/um),...
      sprintf('\\sigma=%5.2f\\mum (fit to double fiber)',sigmaf(2)/um));
set(hleg,'box','off','fontsize',10)
cmd = 'print -depsc Figure04.eps';
fprintf(1,'%s\n',cmd); eval(cmd);
asciify('MeasuredEdgePSF.txt',y/um, Measured.psf);
asciify(sprintf('%05.2fSigmaGaussian.txt',sigma2/um),y/um, psf2);
asciify(sprintf('%05.2fSigmaGaussian.txt',sigmas/um),y/um, psfs);
asciify(sprintf('%05.2fSigmaGaussian.txt',sigmaf(1)/um),yf/um, psff1);
asciify(sprintf('%05.2fSigmaGaussian.txt',sigmaf(2)/um),yf/um, psff2);
asciify('SimSingleFiberWave.txt',yf/um, SimSingle.wv);
asciify('SimSingleFiberStraight.txt',yf/um, SimSingle.st);
asciify('SimDoubleFiberWave.txt',yf/um, SimDouble.wv);
asciify('SimDoubleFiberStraight.txt',yf/um, SimDouble.st);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Convolve the simulated carbon fiber data with the measured PSF.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
SimDouble.wvbm = Convolve( SimDouble.wv , Measured.psf );
SimSingle.wvbm = Convolve( SimSingle.wv , Measured.psf );
SimDouble.stbm = Convolve( SimDouble.st , Measured.psf );
SimSingle.stbm = Convolve( SimSingle.st , Measured.psf );

asciify('SimSingleFiberWaveMeasuredPSF.txt',y/um, SimSingle.wvbm);
asciify('SimSingleFiberStraightMeasuredPSF.txt',y/um, SimSingle.stbm);
asciify('SimDoubleFiberWaveMeasuredPSF.txt',y/um, SimDouble.wvbm);
asciify('SimDoubleFiberStraightMeasuredPSF.txt',y/um, SimDouble.stbm);

fetchfigure('03/11/2005 LANL Fiber Blurred with Measured PSF');clf
   subplot(122)
   hp = plot(y/mm,MeasuredDouble.d,'g',...
     y/mm,SimDouble.wvbm,'b',...
     y/mm,SimDouble.stbm,'r');
   set(hp,'linewidth',1);
   set(hp(1),'linewidth',2);
   set(gca,'ylim',[0.80 1.1]);
   set(gca,'xlim',Width*[-1 1]/2/mm)
   ht = [ xlabel('y (mm)');
  ylabel('Transmission');
  title('Measured PSF, Double Fiber')];
   set(ht,'fontsize',14)
   set(gca,'fontsize',14)
   hleg = legend('Measured','Wave','Straight Ray','location','SouthEast');
   set(hleg,'box','off','fontsize',10)

   subplot(121)
   hp = plot(y/mm,MeasuredSingle.d,'g',...
     y/mm,SimSingle.wvbm,'b',...
     y/mm,SimSingle.stbm,'r');
   set(hp,'linewidth',1);
   set(hp(1),'linewidth',2);
   set(gca,'ylim',[0.80 1.1]);
   set(gca,'xlim',Width*[-1 1]/2/mm)
   ht = [ xlabel('y (mm)');
  ylabel('Transmission');
  title('Measured PSF, Single Fiber')];
   set(ht,'fontsize',14)
   set(gca,'fontsize',14)
   hleg = legend('Measured','Wave','Straight Ray','location','SouthEast');
   set(hleg,'box','off','fontsize',10)

cmd = 'print -depsc Figure05.eps';
fprintf(1,'%s\n',cmd); eval(cmd);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Convolve the simulated carbon fiber data with the measured PSF.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
SimDouble.wvb2 = Convolve( SimDouble.wv , psf2 );
SimSingle.wvb2 = Convolve( SimSingle.wv , psf2 );
SimDouble.stb2 = Convolve( SimDouble.st , psf2 );
SimSingle.stb2 = Convolve( SimSingle.st , psf2 );

asciify(sprintf('SimSingleFiberWave%05.2fPSF.txt',sigma2/um),y/um, SimSingle.wvb2);
asciify(sprintf('SimSingleFiberStraight%05.2fPSF.txt',sigma2/um),y/um, SimSingle.stb2);
asciify(sprintf('SimDoubleFiberWave%05.2fPSF.txt',sigma2/um),y/um, SimDouble.wvb2);
asciify(sprintf('SimDoubleFiberStraight%05.2fPSF.txt',sigma2/um),y/um, SimDouble.stb2);


fetchfigure('03/11/2005 LANL Fiber Blurred with Fit to Measured PSF');clf
   subplot(122)
   hp = plot(y/mm,MeasuredDouble.d,'g',...
     y/mm,SimDouble.wvb2,'b',...
     y/mm,SimDouble.stb2,'r');
   set(hp,'linewidth',1);
   set(hp(1),'linewidth',2);
   set(gca,'ylim',[0.80 1.1]);
   set(gca,'xlim',Width*[-1 1]/2/mm)
   ht = [ xlabel('y (mm)');
  ylabel('Transmission');
  title('Fit to Measured PSF, Double Fiber')];
   set(ht,'fontsize',14)
   set(gca,'fontsize',14)
   hleg = legend('Measured','Wave','Straight Ray','location','SouthEast');
   set(hleg,'box','off','fontsize',10)

   subplot(121)
   hp = plot(y/mm,MeasuredSingle.d,'g',...
     y/mm,SimSingle.wvb2,'b',...
     y/mm,SimSingle.stb2,'r');
   set(hp,'linewidth',1);
   set(hp(1),'linewidth',2);
   set(gca,'ylim',[0.80 1.1]);
   set(gca,'xlim',Width*[-1 1]/2/mm)
   ht = [ xlabel('y (mm)');
  ylabel('Transmission');
  title('Fit to Measured PSF, Single Fiber')];
   set(ht,'fontsize',14)
   set(gca,'fontsize',14)
   hleg = legend('Measured','Wave','Straight Ray','location','SouthEast');
   set(hleg,'box','off','fontsize',10)

cmd = 'print -depsc Figure06.eps';
fprintf(1,'%s\n',cmd); eval(cmd);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Convolve the simulated carbon fiber data with the PSFs fit to the
% measured carbon fiber data.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
SimDouble.wvbf2 = Convolve( SimDouble.wv , psff2 );
SimSingle.wvbf1 = Convolve( SimSingle.wv , psff1 );
SimDouble.stbf2 = Convolve( SimDouble.st , psff2 );
SimSingle.stbf1 = Convolve( SimSingle.st , psff1 );

asciify(sprintf('SimSingleFiberWave%05.2fPSF.txt',sigmaf(1)/um),y/um, SimSingle.wvbf1);
asciify(sprintf('SimSingleFiberStraight%05.2fPSF.txt',sigmaf(1)/um),y/um, SimSingle.stbf1);
asciify(sprintf('SimDoubleFiberWave%05.2fPSF.txt',sigmaf(2)/um),y/um, SimDouble.wvbf2);
asciify(sprintf('SimDoubleFiberStraight%05.2fPSF.txt',sigmaf(2)/um),y/um, SimDouble.stbf2);

fetchfigure('03/11/2005 LANL Fiber Blurred with Fit to Measured Fiber Data');clf
   subplot(122)
   hp = plot(y/mm,MeasuredDouble.d,'g',...
     y/mm,SimDouble.wvbf2,'b',...
     y/mm,SimDouble.stbf2,'r');
   set(hp,'linewidth',1);
   set(hp(1),'linewidth',2);
   set(gca,'ylim',[0.80 1.1]);
   set(gca,'xlim',Width*[-1 1]/2/mm)
   ht = [ xlabel('y (mm)');
  ylabel('Transmission');
  title('Fit to Measured Data, Double Fiber')];
   set(ht,'fontsize',14)
   set(gca,'fontsize',14)
   hleg = legend('Measured','Wave','Straight Ray','location','SouthEast');
   set(hleg,'box','off','fontsize',10)

   subplot(121)
   hp = plot(y/mm,MeasuredSingle.d,'g',...
     y/mm,SimSingle.wvbf1,'b',...
     y/mm,SimSingle.stbf1,'r');
   set(hp,'linewidth',1);
   set(hp(1),'linewidth',2);
   set(gca,'ylim',[0.80 1.1]);
   set(gca,'xlim',Width*[-1 1]/2/mm)
   ht = [ xlabel('y (mm)');
  ylabel('Transmission');
  title('Fit to Measured Data, Single Fiber')];
   set(ht,'fontsize',14)
   set(gca,'fontsize',14)
   hleg = legend('Measured','Wave','Straight Ray','location','SouthEast');
   set(hleg,'box','off','fontsize',10)

cmd = 'print -depsc Figure07.eps';
fprintf(1,'%s\n',cmd); eval(cmd);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Convolve the simulated carbon fiber data with Stefan’s PSF.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
SimDouble.wvbs = Convolve( SimDouble.wv , psfs );
SimSingle.wvbs = Convolve( SimSingle.wv , psfs );
SimDouble.stbs = Convolve( SimDouble.st , psfs );
SimSingle.stbs = Convolve( SimSingle.st , psfs );

fetchfigure(’03/11/2005 LANL Fiber Blurred with Stefan’’s PSF’);clf
   subplot(122)
   hp = plot(y/mm,MeasuredDouble.d,’g’,...
     y/mm,SimDouble.wvbs,’b’,...
     y/mm,SimDouble.stbs,’r’);
   set(hp,’linewidth’,1);
   set(hp(1),’linewidth’,2);
   set(gca,’ylim’,[0.80 1.1]);
   set(gca,’xlim’,Width*[-1 1]/2/mm)
   ht = [ xlabel(’y (mm)’);
  ylabel(’Transmission’);
  title(’Stefan’’s \sigma, Double Fiber’)];
   set(ht,’fontsize’,14)
   set(gca,’fontsize’,14)
   hleg = legend(’Measured’,’Wave’,’Straight Ray’,’location’,’SouthEast’);
   set(hleg,’box’,’off’,’fontsize’,10)

   subplot(121)
   hp = plot(y/mm,MeasuredSingle.d,’g’,...
     y/mm,SimSingle.wvbs,’b’,...
     y/mm,SimSingle.stbs,’r’);
   set(hp,’linewidth’,1);
   set(hp(1),’linewidth’,2);
   set(gca,’ylim’,[0.80 1.1]);
   set(gca,’xlim’,Width*[-1 1]/2/mm)
   ht = [ xlabel(’y (mm)’);
  ylabel(’Transmission’);
  title(’Stefan’’s \sigma, Single Fiber’)];
   set(ht,’fontsize’,14)
   set(gca,’fontsize’,14)
   hleg = legend(’Measured’,’Wave’,’Straight Ray’,’location’,’SouthEast’);
   set(hleg,’box’,’off’,’fontsize’,10)

cmd = ’print -depsc Figure08.eps’;
fprintf(1,’%s\n’,cmd); eval(cmd);
```

C.4 Code to Process the Data Sets II

This code, `Main2.m`, fits the data to a sum of two Gaussians.

```
%*****************************************************************************
%
% TITLE:    Main2.m
% AUTHOR:   Sean K. Lehman
% DATE:     September 20, 2005
% FUNCTION: Process the data from the
%               AuCuEdge/2005.06.23.LANL
%           and
%               ThreeCarbonWires/2005.03.11.LANL
%           directories
%           Fit double Gaussian PSFs.
% SYNTAX:
% CALLS:
%
% MODIFICATIONS:
```

```
%
%
%    (c) Copyright 2005 the Regents of the University
%             of California. All rights reserved.
%
%     This work was produced at the Lawrence Livermore
%     National Laboratory. The United States Government
%     retains certain rights therein.
%
%***********************************************************************/
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% AuCuEdge/2005.06.23.LANL/LANL.mat contains
% Measured =
%      y: [273x1 double]
%      d: [273x1 double]
%    psf: [273x1 double]
%     dy: 3.6600e-07
%
% Simulated683 =
%      y: [273x1 double]
%     wv: [273x1 double]
%     st: [273x1 double]
%     dy: 3.6600e-07
%
% Simulated716 =
%      y: [273x1 double]
%     wv: [273x1 double]
%     st: [273x1 double]
%     dy: 3.6600e-07
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
load ../AuCuEdge/2005.06.23.LANL/LANL.mat
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ThreeCarbonWires/2005.03.11.LANL/LANL.03.11.2005.mat contains
% MeasuredDouble =
%      y: [273x1 double]
%      d: [273x1 double]
%     dy: 3.6700e-07
%
% MeasuredSingle =
%      y: [273x1 double]
%      d: [273x1 double]
%     dy: 3.6700e-07
%
% SimDouble =
%     name: '3/11/2005 LANL, 0 rotation, double wire'
%        y: [273x1 double]
%       st: [273x1 double]
%       wv: [273x1 double]
%       dy: 3.6700e-07
%
% SimSingle =
%     name: '3/11/2005 LANL, 0 rotation, single wire'
%        y: [273x1 double]
%       st: [273x1 double]
%       wv: [273x1 double]
%       dy: 3.6700e-07
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
load ../ThreeCarbonWires/2005.03.11.LANL/LANL.03.11.2005.mat
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Preliminary definitions
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
um    = 1e-6;
mm    = 1e-3;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Range to be plotted
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Width = 0.06 * mm;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Fit the data using double Gaussians
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
sigma0 = 2.5 * um;
alpha0 = 1;
alpha0 = 0.06;
omega0 = [ alpha0 alpha0 sigma0 sigma0/10 -um um ];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Tinker with options
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
opts   = optimset(optimset(’fminsearch’),...
  ’tolx’,1e-10,...
  ’tolfun’,1e-10,...
  ’MaxIter’,    500*length(omega0),...
  ’MaxFunEvals’,800*length(omega0));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Solve for the Gaussian fit
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
y    = Measured.y;
md   = Measured.d;
psfm = Measured.psf;
sd   = Simulated683.wv;
dy   = Measured.dy;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Sigma1 is fit by blurring the edge data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[omega1,fval,iflag,output]=fminsearch(@(o)FitLANLEdge2(o,y,md,sd),omega0,opts);
fprintf(1,’################################################\n’);
fprintf(1,’%s\n’,output.message);
fprintf(1,’################################################\n’);
Ng    = length( omega1 ) / 3;
% omega = [ alpha alpha ... sigma sigma ... x0 x0 ... ];
alpha1 = omega1(1:Ng);
sigma1 = omega1(Ng+1:2*Ng);
x01    = omega1(2*Ng+1:3*Ng);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Fit the data using double Gaussians
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
sigma0 = 2.96 * um;
alpha0 = 0.05;
omega0 = [ alpha0 2*alpha0 sigma0 sigma0/10 -um um ];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Sigma2 is fit directly to the PSF. There is a difference.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[omega2,fval,iflag,output]=fminsearch(@(o)FitLANLPSF2(o,y,psfm),omega0,opts);
fprintf(1,’################################################\n’);
fprintf(1,’%s\n’,output.message);
fprintf(1,’################################################\n’);
Ng    = length( omega1 ) / 3;
% omega = [ alpha alpha ... sigma sigma ... x0 x0 ... ];
alpha2 = omega2(1:Ng);
sigma2 = omega2(Ng+1:2*Ng);
x02    = omega2(2*Ng+1:3*Ng);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Define Stefan’s sigma
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
sigmas = 3.4 * um;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Print results
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(1,’fval  : %g\n’,fval);
fprintf(1,'iflag : %d\n',iflag);
fprintf(1,'Output Message: %s\n',output.message);

fprintf(1,'sigma1 = ');fprintf(1,'%g ',sigma1/um);fprintf(1,'micrometers\n');
fprintf(1,'sigma2 = ');fprintf(1,'%g ',sigma2/um);fprintf(1,'micrometers\n');

fprintf(1,'alpha1 = ');fprintf(1,'%g ',alpha1);fprintf(1,'\n');
fprintf(1,'alpha2 = ');fprintf(1,'%g ',alpha2);fprintf(1,'\n');

fprintf(1,'x01    = ');fprintf(1,'%g ',x01/um);fprintf(1,'micrometers\n');
fprintf(1,'x02    = ');fprintf(1,'%g ',x02/um);fprintf(1,'micrometers\n');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Compute the PSFs
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
psf1  = multigaussian(y,omega1);
psf1  = psf1/sum(psf1);
psf2  = multigaussian(y,omega2);
psf2  = psf2/sum(psf2);
psfs  = multigaussian(y,[],sigmas,0);
psfs  = psfs/sum(psfs);

[fwhm1 hm1]  = FullWidthHalfMax(y/mm,psf1);
[fwhm2 hm2]  = FullWidthHalfMax(y/mm,psf2);
[fwhm3 hm3]  = FullWidthHalfMax(y/mm,psfs);
[fwhm4 hm4]  = FullWidthHalfMax(y/mm,Measured.psf);
fwhm         = [
%    diff(fwhm1)*mm/um
    diff(fwhm2)*mm/um
    diff(fwhm3)*mm/um
    diff(fwhm4)*mm/um
       ];
hm           = [ hm1 hm2 hm3 hm4 ];
hm           = [     hm2 hm3 hm4 ];
pstr = {
%    [sprintf('\\sigma1 = ') sprintf('%5.2f ',sigma1/um) sprintf('\\mum')];
    [sprintf('\\sigma = ') sprintf('%5.2f ',sigma2/um) sprintf('\\mum')];
%    [sprintf('\\alpha1 = ') sprintf('%5.2f ',alpha1) ];
    [sprintf('\\alpha = ') sprintf('%5.2f ',alpha2) ];
%    [sprintf('x01 = ') sprintf('%5.2f ',x01/um) sprintf('\\mum')];
    [sprintf('x0 = ') sprintf('%5.2f ',x02/um) sprintf('\\mum')];
    [sprintf('fwhm = ') sprintf('%5.2f ',fwhm) sprintf('\\mum')];
    [sprintf('hm = ') sprintf('%5.2f ',hm)];
       };

str = 'LANL Au/Cu Edge One Gaussian PSFs';
cmd = sprintf('fetchfigure(''%s'');clf',str);
fprintf(1,'%s\n',cmd); eval( cmd );

if 1
  hp = plot(y/um,psfm,'g',...
    y/um,psf2,'b',...
    y/um,psfs,'r');
else
  hp = plot(y/um,psfm,'g',...
    y/um,psf1,'k',...
    y/um,psf2,'b',...
    y/um,psfs,'r');
end

set(gca,'xlim',Width*[-1 1]/4/um)
set(hp,'linewidth',1);set(hp(1),'linewidth',2)
ht = [ title(str); xlabel('y (\mum)') ];
set(ht,'fontsize',14)
set(gca,'fontsize',14)
```

```
xlim  = get(gca,’xlim’);  Dxlim = diff(xlim);
ylim  = get(gca,’ylim’);  Dylim = diff(ylim);
dytxt = 0.1 * Dylim;
for m=1:length(pstr)
  htxt = text(xlim(1)+0.025*Dxlim,ylim(2)-m*dytxt,pstr(m));
  set(htxt,’fontsize’,12);
end
if 1
  hleg = legend(’Measured’,...
’Fit to measured PSF’,...
sprintf(’\\sigma=%5.2f\\mum (Stefan)’,sigmas/um));
else
  hleg = legend(’Measured’,...
’Fit to blurred data’,...
’Fit to measured PSF’,...
sprintf(’\\sigma=%5.2f\\mum (Stefan)’,sigmas/um));
end

set(hleg,’box’,’off’,’fontsize’,10)

cmd = ’print -depsc Figure01.2.eps’;
fprintf(1,’%s\n’,cmd); eval(cmd);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Blur the simulated edge data with the omega2 multi-Gaussian fit
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Simulated683.wvb = ApplyMultiGaussianBlur(y,Simulated683.wv,omega2);
Simulated683.stb = ApplyMultiGaussianBlur(y,Simulated683.st,omega2);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Plot edge blurred data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fetchfigure(’LANL Au/Cu Blurred’);clf
   hp = plot(y/mm,Measured.d,’g’,...
     y/mm,Simulated683.wvb,’b’,...
     y/mm,Simulated683.stb,’r--’);
   set(hp,’linewidth’,1);
   set(hp(1),’linewidth’,2);
   axis tight
   set(gca,’xlim’,Width*[-1 1]/2/mm)
   ht = [ xlabel(’y (mm)’);
  ylabel(’Transmission’);
  title(’Blurred Simulated Au/Cu Edge’)];
   set(ht,’fontsize’,14)
   set(gca,’fontsize’,14)
   hleg = legend(’Measured’,...
 ’Wave Simulation’,...
 ’Straigt Ray Simulation’,’location’,’SouthEast’);

cmd = ’print -depsc Figure02.2.eps’;
fprintf(1,’%s\n’,cmd); eval(cmd);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Now start working with the carbon fiber data.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(1,’Measured single dy : %g micrometers\n’,MeasuredSingle.dy/um);
fprintf(1,’Measured double dy : %g micrometers\n’,MeasuredDouble.dy/um);
fprintf(1,’Simulated single dy: %g micrometers\n’,SimSingle.dy/um);
fprintf(1,’Simulated double dy: %g micrometers\n’,SimDouble.dy/um);

yf   = MeasuredDouble.y;
dyf  = MeasuredDouble.dy;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Measured data single wire
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
mds = MeasuredSingle.d;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Measured data double wire
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
mdd = MeasuredDouble.d;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Simulated data single wire
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
sds = SimSingle.wv;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Simulated data double wire
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
sdd = SimDouble.wv;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Fit the data.
% Fit the single and double wires separately
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Values for initial guess
sigma0 = [3.90 4.28]*um;
alpha0 = 0.035;
omega0 = [
    alpha0 alpha0 sigma0(1) sigma0(1) -um um ...
    alpha0 alpha0 sigma0(2) sigma0(2) -um um
 ];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Tinker with options
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
opts       = optimset(optimset('fminsearch'),...
      'tolx',1e-10,...
      'tolfun',1e-10,...
      'MaxIter',    600*length(omega0),...
      'MaxFunEvals',800*length(omega0));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Solve for the Gaussian fit
[omegaf,fval,iflag,output]=fminsearch(@(o)FitLANLFiber2(o,yf,mds,mdd,sds,sdd),...
      omega0,opts);
fprintf(1,'################################################\n');
fprintf(1,'%s\n',output.message);
fprintf(1,'################################################\n');
No      = length( omegaf );
omegaf1 = omegaf( 1:No/2 );
omegaf2 = omegaf( No/2+1:No );
Ng      = length( omegaf1 ) / 3;
% omega = [ alpha alpha ... sigma sigma ... x0 x0 ... ];
alphaf1 = omegaf1(1:Ng);
sigmaf1 = omegaf1(Ng+1:2*Ng);
x0f1    = omegaf1(2*Ng+1:3*Ng);
alphaf2 = omegaf2(1:Ng);
sigmaf2 = omegaf2(Ng+1:2*Ng);
x0f2    = omegaf2(2*Ng+1:3*Ng);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf(1,'fval  : %g\n',fval);
fprintf(1,'iflag : %d\n',iflag);
fprintf(1,'Output Message: %s\n',output.message);

fprintf(1,'sigmaf1 = ');fprintf(1,'%g ',sigmaf1/um);fprintf(1,'micrometers\n');
fprintf(1,'sigmaf2 = ');fprintf(1,'%g ',sigmaf2/um);fprintf(1,'micrometers\n');

fprintf(1,'alphaf1 = ');fprintf(1,'%g ',alphaf1);fprintf(1,'\n');
fprintf(1,'alphaf2 = ');fprintf(1,'%g ',alphaf2);fprintf(1,'\n');

fprintf(1,'x0f1    = ');fprintf(1,'%g ',x0f1/um);fprintf(1,'micrometers\n');
fprintf(1,'x0f2    = ');fprintf(1,'%g ',x0f2/um);fprintf(1,'micrometers\n');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Compute the fitted PSF
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
psff1 = multigaussian(yf,omegaf1);
psff1 = psff1 / sum( psff1 );
psff2 = multigaussian(yf,omegaf2);
psff2 = psff2 / sum( psff2 );

[fwhm1 hm1] = FullWidthHalfMax(yf/mm,psff1);
[fwhm2 hm2] = FullWidthHalfMax(yf/mm,psff2);
fwhm        = [diff(fwhm1)*mm/um diff(fwhm2)*mm/um];
hm          = [ hm1 hm2 ];
pstr = {
    [sprintf('\\sigma1 = ') sprintf('%5.2f ',sigmaf1/um) sprintf('\\mum')];
    [sprintf('\\sigma2 = ') sprintf('%5.2f ',sigmaf2/um) sprintf('\\mum')];
    [sprintf('\\alpha1 = ') sprintf('%5.2f ',alphaf1)];
    [sprintf('\\alpha2 = ') sprintf('%5.2f ',alphaf2)];
    [sprintf('x01 = ') sprintf('%5.2f ',x0f1/um) sprintf('\\mum')];
    [sprintf('x02 = ') sprintf('%5.2f ',x0f2/um) sprintf('\\mum')];
    [sprintf('fwhm = ') sprintf('%5.2f ',fwhm) sprintf('\\mum')];
    [sprintf('hm = ') sprintf('%5.2f ',hm)];
       };

str = 'March 11, 2005, LANL Separate Two Gaussian PSFs';
cmd = sprintf('fetchfigure(''%s'');clf',str);
fprintf(1,'%s\n',cmd); eval( cmd );

hp = plot(yf/mm,psff1,'b',yf/mm,psff2,'r');
set(gca,'xlim',Width*[-1 1]/4/mm)
set(hp,'linewidth',1)
ht = [ title(str); xlabel('y (mm)') ];
set(ht,'fontsize',14)
set(gca,'fontsize',14)

xlim  = get(gca,'xlim');  Dxlim = diff(xlim);
ylim  = get(gca,'ylim');  Dylim = diff(ylim);
dytxt = 0.1 * Dylim;
for m=1:length(pstr)
  htxt = text(xlim(1)+0.025*Dxlim,ylim(2)-m*dytxt,pstr(m));
  set(htxt,'fontsize',12);
end
legend('Single Fiber','Double Fiber');
cmd = 'print -depsc Figure03.2.eps';
fprintf(1,'%s\n',cmd); eval(cmd);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Plot all the PSFs together
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fetchfigure('All PSFs');clf
hp = plot(y/mm , psfm,'g',...
  y/mm , psf2,'k',...
  y/mm , psfs,'r',...
  yf/mm, psff1,'b',...
  yf/mm, psff2,'m');
set(gca,'xlim',Width*[-1 1]/4/mm)
set(hp,'linewidth',1);set(hp(1),'linewidth',2)
ht = [ title('Comparison of all PSFs'); xlabel('y (mm)') ];
set(ht,'fontsize',14)
set(gca,'fontsize',14)

hleg = legend('Measured Edge',...
      sprintf('\\sigma=[%5.2f %5.2f]\\mum (Fit to measured PSF)',...
      sigma2(1)/um,sigma2(2)/um),...
      sprintf('\\sigma=%5.2f\\mum (Stefan)',sigmas/um),...
      sprintf('\\sigma=[%5.2f %5.2f]\\mum (Fit to single fiber)',...
      sigmaf1(1)/um,sigmaf1(2)/um),...
      sprintf('\\sigma=[%5.2f %5.2f]\\mum (Fit to double fiber)',...
      sigmaf2(1)/um,sigmaf2(2)/um));
set(hleg,'box','off','fontsize',10)
cmd = 'print -depsc Figure04.2.eps';
fprintf(1,'%s\n',cmd); eval(cmd);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Convolve the simulated carbon fiber data with the measured PSF.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
SimDouble.wvb2 = Convolve( SimDouble.wv , psf2 );
SimSingle.wvb2 = Convolve( SimSingle.wv , psf2 );
SimDouble.stb2 = Convolve( SimDouble.st , psf2 );
SimSingle.stb2 = Convolve( SimSingle.st , psf2 );

fetchfigure('03/11/2005 LANL Fiber Blurred with Fit to Measured PSF');clf
   subplot(122)
   hp = plot(y/mm,MeasuredDouble.d,'g',...
     y/mm,SimDouble.wvb2,'b',...
     y/mm,SimDouble.stb2,'r');
   set(hp,'linewidth',1);
   set(hp(1),'linewidth',2);
   set(gca,'ylim',[0.80 1.1]);
   set(gca,'xlim',Width*[-1 1]/2/mm)
   ht = [ xlabel('y (mm)');
  ylabel('Transmission');
  title('Fit to Measured PSF, Double Fiber')];
   set(ht,'fontsize',14)
   set(gca,'fontsize',14)
   hleg = legend('Measured','Wave','Straight Ray','location','SouthEast');
   set(hleg,'box','off','fontsize',10)

   subplot(121)
   hp = plot(y/mm,MeasuredSingle.d,'g',...
     y/mm,SimSingle.wvb2,'b',...
     y/mm,SimSingle.stb2,'r');
   set(hp,'linewidth',1);
   set(hp(1),'linewidth',2);
   set(gca,'ylim',[0.80 1.1]);
   set(gca,'xlim',Width*[-1 1]/2/mm)
   ht = [ xlabel('y (mm)');
  ylabel('Transmission');
  title('Fit to Measured PSF, Single Fiber')];
   set(ht,'fontsize',14)
   set(gca,'fontsize',14)
   hleg = legend('Measured','Wave','Straight Ray','location','SouthEast');
   set(hleg,'box','off','fontsize',10)

cmd = 'print -depsc Figure06.2.eps';
fprintf(1,'%s\n',cmd); eval(cmd);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Convolve the simulated carbon fiber data with the PSFs fit to the
% measured carbon fiber data.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
SimDouble.wvbf2 = Convolve( SimDouble.wv , psff2 );
SimSingle.wvbf1 = Convolve( SimSingle.wv , psff1 );
SimDouble.stbf2 = Convolve( SimDouble.st , psff2 );
SimSingle.stbf1 = Convolve( SimSingle.st , psff1 );

fetchfigure('03/11/2005 LANL Fiber Blurred with Fit to Measured Fiber Data');clf
   subplot(122)
   hp = plot(y/mm,MeasuredDouble.d,'g',...
     y/mm,SimDouble.wvbf2,'b',...
     y/mm,SimDouble.stbf2,'r');
   set(hp,'linewidth',1);
   set(hp(1),'linewidth',2);
   set(gca,'ylim',[0.80 1.1]);
   set(gca,'xlim',Width*[-1 1]/2/mm)
   ht = [ xlabel('y (mm)');
  ylabel('Transmission');
  title('Fit to Measured Data, Double Fiber')];
   set(ht,'fontsize',14)
   set(gca,'fontsize',14)
   hleg = legend('Measured','Wave','Straight Ray','location','SouthEast');
   set(hleg,'box','off','fontsize',10)

   subplot(121)
   hp = plot(y/mm,MeasuredSingle.d,'g',...
     y/mm,SimSingle.wvbf1,'b',...
     y/mm,SimSingle.stbf1,'r');
   set(hp,'linewidth',1);
   set(hp(1),'linewidth',2);
   set(gca,'ylim',[0.80 1.1]);
   set(gca,'xlim',Width*[-1 1]/2/mm)
   ht = [ xlabel('y (mm)');
  ylabel('Transmission');
  title('Fit to Measured Data, Single Fiber')];
   set(ht,'fontsize',14)
   set(gca,'fontsize',14)
   hleg = legend('Measured','Wave','Straight Ray','location','SouthEast');
   set(hleg,'box','off','fontsize',10)

cmd = 'print -depsc Figure07.2.eps';
fprintf(1,'%s\n',cmd); eval(cmd);
```

C.5 Apply a Multi-Gaussian PSF to a Data set

```
function [ db , g ] = ApplyMultiGaussianBlur( y , d , omega )
%****************************************************************************
%
% TITLE:    ApplyMultiGaussianBlur.m
% AUTHOR:   Sean K. Lehman
% DATE:     June 28, 2005
% FUNCTION: Apply a multi-Gaussian blur to the simulated data.
% SYNTAX:
%
% MODIFICATIONS:
%
%
%    (c) Copyright 2005 the Regents of the University
%             of California. All rights reserved.
%
%     This work was produced at the Lawrence Livermore
%     National Laboratory. The United States Government
%     retains certain rights therein.
%
%***************************************************************************/
% Ensure the y variable is centered about zero
   y     = y - mean( y );

% Sample interval
   dy    = y(2) - y(1);

% Construct the multigaussian
   g = multigaussian(y,omega);
   g = g / sum( g );
%   g = g / ( dy * sum( g ) );
% Apply the blur
   Ny    = length( y );
   mud   = mean( d );
   d     = d - mud;
   Nconv = 2*Ny - 1;
   N1    = fix( (Nconv-Ny) / 2 );
   N2    = N1 + Ny - 1;
   db    = conv( d , g );
   db    = db(N1:N2) + mud;
```

C.6 Convolution Routine

```
function db = Convolve( d , g )
%****************************************************************************
%
% TITLE:    Convolve.m
% AUTHOR:   Sean K. Lehman
% DATE:     September 20, 2005
% FUNCTION:
% SYNTAX:
% CALLS:
%
% MODIFICATIONS:
%
%
%    (c) Copyright 2005 the Regents of the University
%             of California. All rights reserved.
%
%     This work was produced at the Lawrence Livermore
%     National Laboratory. The United States Government
%     retains certain rights therein.
%
%***************************************************************************/
% Get lengths
   Nd    = length( d );
   Ng    = length( g );
   Ny    = min([ Nd Ng ]);
% Apply the blur
   mud   = mean( d );
   d     = d - mud;
   Nconv = Nd + Ng - 1;
   N1    = fix( (Nconv-Ny) / 2 );
   N2    = N1 + Ny - 1;
   db    = conv( d , g );
   db    = db(N1:N2) + mud;
```

C.7 Multi-Gaussian Function Generator

```
function [f g] = multigaussian(x,alpha,sigma,x0)
%****************************************************************************
%
% TITLE:    multigaussian.m
% AUTHOR:   Sean K. Lehman
% DATE:     June 27, 2005
% FUNCTION: f     = multigaussian(x,alpha,sigma,x0)
%           [f g] = multigaussian(x,alpha,sigma,x0)
%           f     = multigaussian(x,omega)
%           [f g] = multigaussian(x,omega)
% SYNTAX:
%
```

```
% MODIFICATIONS:
%
%
%    (c) Copyright 2005 the Regents of the University
%             of California. All rights reserved.
%
%     This work was produced at the Lawrence Livermore
%     National Laboratory. The United States Government
%     retains certain rights therein.
%
%*************************************************************************/
Normalize = 0;

if nargin == 2
  omega = alpha;
  % Number of Gaussians:
  Ng    = length( omega ) / 3;
  % omega = [ alpha alpha ... sigma sigma ... x0 x0 ... ];
  alpha = omega(1:Ng);
  sigma = omega(Ng+1:2*Ng);
  x0    = omega(2*Ng+1:3*Ng);
else
  % Number of Gaussians:
  if isempty( alpha )
    Normalize = 1;
    Ng        = length( sigma );
  else
    Ng    = length( alpha );
  end
end

if Ng ~= length( sigma )
  fprintf(1,'Number of amplitudes differs from the number of widths\n');
  return
end

if Ng ~= length( x0 )
  fprintf(1,'Number of amplitudes differs from the number of shifts\n');
  return
end

Nx  = length( x );
g   = zeros( [ Nx Ng ] );


% Construct the multigaussian
if Normalize
  for n = 1:Ng
    arg    = ((x-x0(n))/sigma(n)).^2 ;
    g(:,n) = exp(-arg/2);
  end
  f = sum( g , 2 );
  f = f / sum(f);
else
  for n = 1:Ng
    arg    = ((x-x0(n))/sigma(n)).^2 ;
    g(:,n) = alpha(n) * exp(-arg/2);
  end
  f = sum( g , 2 );
end


if 0
  fetchfigure('multigaussian');clf
  plot(x,f)
end
```

C.8 LMS Error Function to Fit a Single Gaussian to the Edge Data via Convolution

```
function err = FitLANLEdge(sigma,y,md,sd)
%*****************************************************************************
%
% TITLE:    FitLANLEdge.m
% AUTHOR:   Sean K. Lehman
% DATE:     September 20, 2005
% FUNCTION: Error function called by optimization routine in Main.m
% SYNTAX:
% CALLS:
%
% MODIFICATIONS:
%
%
%    (c) Copyright 2005 the Regents of the University
%             of California. All rights reserved.
%
%     This work was produced at the Lawrence Livermore
%     National Laboratory. The United States Government
%     retains certain rights therein.
%
%***************************************************************************/
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Function called by fminsearch
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
arg    = (y/sigma).^2;
f      = exp(-arg/2);
omega  = [ sum(f) sigma 0 ];

b      = ApplyMultiGaussianBlur( y , sd , omega );
model  = b;
data   = md;
err    = sqrt( sum( (model - data).^2 ) );
```

C.9 LMS Error Function to Fit a Double Gaussian to the Edge Data via Convolution

```
function err = FitLANLEdge2(omega,y,md,sd)
%*****************************************************************************
%
% TITLE:    FitLANLEdge2.m
% AUTHOR:   Sean K. Lehman
% DATE:     September 20, 2005
% FUNCTION: Error function called by optimization routine in Main.m
% SYNTAX:
% CALLS:
%
% MODIFICATIONS:
%
%
%    (c) Copyright 2005 the Regents of the University
%             of California. All rights reserved.
%
%     This work was produced at the Lawrence Livermore
%     National Laboratory. The United States Government
%     retains certain rights therein.
```

```
%
%*************************************************************************/
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Function called by fminsearch
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
b      = ApplyMultiGaussianBlur( y , sd , omega );
model  = b;
data   = md;
err    = sqrt( sum( (model - data).^2 ) );
```

C.10 LMS Error Function to Fit a Single Gaussian to the Carbon Fiber Data via Convolution

```
function err = FitLANLFiber(sigma,y,mds,mdd,sds,sdd)
%***************************************************************************
%
% TITLE:    FitLANLFiber.m
% AUTHOR:   Sean K. Lehman
% DATE:     September 20, 2005
% FUNCTION: Error function called by optimization routine in Main.m
%           We fit the single and double wire separately
% SYNTAX:
% CALLS:
%
% MODIFICATIONS:
%
%
%    (c) Copyright 2005 the Regents of the University
%             of California. All rights reserved.
%
%     This work was produced at the Lawrence Livermore
%     National Laboratory. The United States Government
%     retains certain rights therein.
%
%*************************************************************************/
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Function called by fminsearch
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
arg1   = (y/sigma(1)).^2;
arg2   = (y/sigma(2)).^2;
f1     = exp(-arg1/2);
f2     = exp(-arg2/2);
omega1 = [ sum(f1) sigma(1) 0 ];
omega2 = [ sum(f2) sigma(2) 0 ];

b1     = ApplyMultiGaussianBlur( y , sds , omega1 );
b2     = ApplyMultiGaussianBlur( y , sdd , omega2 );
model  = [ b1 ; b2 ];
data   = [ mds ; mdd ];
err    = sqrt( sum( (model - data).^2 ) );
```

C.11 LMS Error Function to Fit a Double Gaussian to the Carbon Fiber Data via Convolution

```
function err = FitLANLFiber2(omega,y,mds,mdd,sds,sdd)
%***************************************************************************
%
% TITLE:    FitLANLFiber2.m
% AUTHOR:   Sean K. Lehman
% DATE:     September 20, 2005
% FUNCTION: Error function called by optimization routine in Main.m
%           We fit the single and double wire separately
% SYNTAX:
% CALLS:
%
% MODIFICATIONS:
%
%
%    (c) Copyright 2005 the Regents of the University
%             of California. All rights reserved.
%
%     This work was produced at the Lawrence Livermore
%     National Laboratory. The United States Government
%     retains certain rights therein.
%
%***************************************************************************/
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Function called by fminsearch
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
No     = length( omega );
omega1 = omega( 1:No/2 );
omega2 = omega( No/2+1:No );

b1     = ApplyMultiGaussianBlur( y , sds , omega1 );
b2     = ApplyMultiGaussianBlur( y , sdd , omega2 );
model  = [ b1 ; b2 ];
data   = [ mds ; mdd ];
err    = sqrt( sum( (model - data).^2 ) );
```

C.12 LMS Error Function to Fit a Single Gaussian to the Measured Edge PSF

```
function err = FitLANLPSF(sigma,y,MeasuredPSF)
%****************************************************************************
%
% TITLE:    FitLANLPSF.m
% AUTHOR:   Sean K. Lehman
% DATE:     September 20, 2005
% FUNCTION: Error function called by optimization routine in Main.m
%           Here we fit the actual PSF and not the blurred data.
% SYNTAX:
% CALLS:
%
% MODIFICATIONS:
%
%
%    (c) Copyright 2005 the Regents of the University
%             of California. All rights reserved.
%
%     This work was produced at the Lawrence Livermore
%     National Laboratory. The United States Government
%     retains certain rights therein.
%
%***************************************************************************/
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Function called by fminsearch
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

model  = multigaussian(y,[],sigma,0);
model  = model / sum( model );
data   = MeasuredPSF;
err    = sqrt( sum( (model - data).^2 ) );
```

C.13 LMS Error Function to Fit a Double Gaussian to the Measured Edge PSF

```
function err = FitLANLPSF2(omega,y,MeasuredPSF)
%*****************************************************************************
%
% TITLE:    FitLANLPSF2.m
% AUTHOR:   Sean K. Lehman
% DATE:     September 20, 2005
% FUNCTION: Error function called by optimization routine in Main.m
%           Here we fit the actual PSF and not the blurred data.
% SYNTAX:
% CALLS:
%
% MODIFICATIONS:
%
%
%    (c) Copyright 2005 the Regents of the University
%             of California. All rights reserved.
%
%     This work was produced at the Lawrence Livermore
%     National Laboratory. The United States Government
%     retains certain rights therein.
%
%***************************************************************************/
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Function called by fminsearch
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

model  = multigaussian(y,omega);
model  = model / sum( model );
data   = MeasuredPSF;
err    = sqrt( sum( (model - data).^2 ) );
```

C.14 Centered Two Point First Derivative Function

```
function fp = fd1( f , dt )
%*****************************************************************************
%
% TITLE:    fd1.m
% AUTHOR:   Sean K. Lehman
% DATE:     December 15, 2004
% FUNCTION: Compute a two point finite difference to approximate
%           the second derivative.
% SYNTAX:
%
% MODIFICATIONS:
%
%
%    (c) Copyright 2004 the Regents of the University
%             of California. All rights reserved.
%
%     This work was produced at the Lawrence Livermore
%     National Laboratory. The United States Government
%     retains certain rights therein.
%
%***************************************************************************/
   Nt    = length( f );
   denom = 2 * dt;
   fp    = ( shift(f,-1) - shift(f,1) ) / denom;
   % Correct for edges of the grid
   loc     = 1;
   fp(loc) = (-3 * f(loc) + 4 * f(loc+1) - f(loc+2)) / denom;
```

```
loc     = Nt;
fp(loc) = ( 3 * f(loc) - 4 * f(loc-1) + f(loc-2)) / denom;
```

References

[1] G. Gbur and E. Wolf. Relation between computed tomography and diffraction tomography. *Journal of the Optical Society of America A*, **18**(9):2132–2137, September 2001.